

CE



Original-Programmierhandbuch ClassicController

ecomat100

CR0033

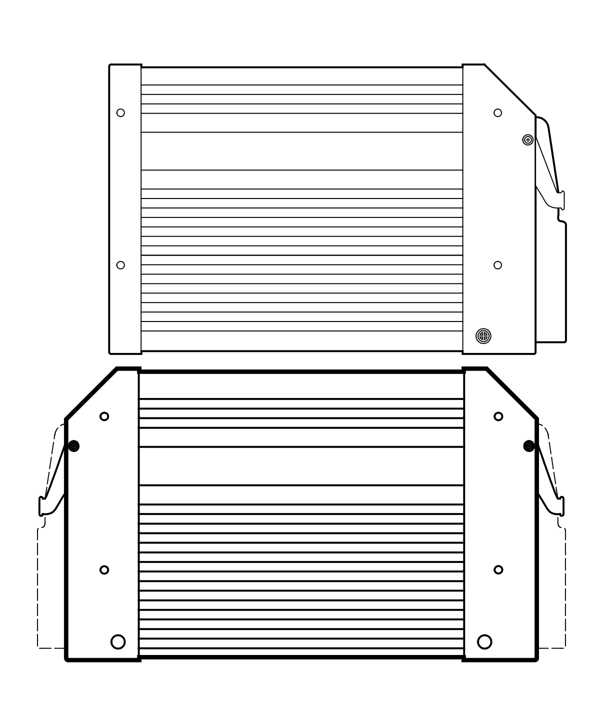
mit integriertem E/A-Modul: **CR0133**

Laufzeitsystem V01.00.09

CODESYS® V2.3

Deutsch

7390969_08_DE 2015-09-30



Inhaltsverzeichnis

1 Über diese Anleitung

Inhalt

202

1.1 Copyright

6088

Alle auf unseren Seiten verwendeten Produktnamen, -Bilder, Unternehmen oder sonstige Marken sind Eigentum der jeweiligen Rechteinhaber:

- AS-i ist Eigentum der AS-International Association, (→ www.as-interface.net)
- CAN ist Eigentum der CiA (CAN in Automation e.V.), Deutschland (→ www.can-cia.org)
- CODESYS™ ist Eigentum der 3S – Smart Software Solutions GmbH, Deutschland (→ www.codesys.com)
- DeviceNet™ ist Eigentum der ODVA™ (Open DeviceNet Vendor Association), USA (→ www.odva.org)
- EtherNet/IP® ist Eigentum der →ODVA™
- IO-Link® (→ www.io-link.com) ist Eigentum der →PROFIBUS Nutzerorganisation e.V., Deutschland
- Microsoft® ist Eigentum der Microsoft Corporation, USA (→ www.microsoft.com)
- PROFIBUS® ist Eigentum der PROFIBUS Nutzerorganisation e.V., Deutschland (→ www.profibus.com)
- PROFINET® ist Eigentum der →PROFIBUS Nutzerorganisation e.V., Deutschland
- Windows® ist Eigentum der →Microsoft Corporation, USA

1.2 Übersicht: Dokumentations-Module für ecomatmobile-Geräte

17405

Die Dokumentation für **ecomat*mobile***-Geräte besteht aus folgenden Modulen:

1.	Datenblatt
Inhalt:	Technische Daten in Tabellenform
Quelle:	→ www.ifm.com > Land wählen > [Datenblattsuche] > CR0033 > [Technische Daten im PDF-Format]

2.	Montageanleitung / Betriebsanleitung
Inhalt:	Anleitung für Montage, elektrische Installation, (Inbetriebnahme*), Technische Daten
Quelle:	Anleitung wird mit dem Gerät mitgeliefert Auch zu finden auf der ifm-Homepage: → www.ifm.com > Land wählen > [Datenblattsuche] > CR0033 > [Betriebsanleitungen]

3.	Programmierhandbuch + Online-Hilfe
Inhalt:	Beschreibung der Konfiguration und der Funktionen der Geräte-Software
Quelle:	→ www.ifm.com > Land wählen > [Datenblattsuche] > CR0033 > [Betriebsanleitungen]

4.	Systemhandbuch "Know-How ecomatmobile"
Inhalt:	Hintergrundwissen zu folgenden Themen: • Übersicht Templates und Demo-Programme • CAN, CANopen • Ausgänge steuern • User-Flash-Speicher • Visualisierungen • Übersicht Dateien und Bibliotheken
Quelle:	→ www.ifm.com > Land wählen > [Datenblattsuche] > CR0033 > [Betriebsanleitungen]

*) Die in Klammern gesetzten Beschreibungen sind nur in den Anleitungen bestimmter Geräte enthalten.

1.3 CODESYS-Programmierhandbuch

17542

Im ergänzenden "Programmierhandbuch CODESYS V2.3" der 3S GmbH erhalten Sie weitergehende Informationen über die Nutzung des Programmiersystems.
Dieses Handbuch steht auf der **ifm**-Homepage als kostenloser Download zur Verfügung:
→ www.ifm.com > Land wählen > [Service] > [Download] > [Systeme für mobile Arbeitsmaschinen]
Handbücher und Online-Hilfen für **ecomat*mobile*** finden Sie auch hier:
→ **ecomat*mobile***-DVD "Software, tools and documentation"

1.4 Was bedeuten die Symbole und Formatierungen?

203

Folgende Symbole oder Piktogramme verdeutlichen Ihnen unsere Hinweise in unseren Anleitungen:

⚠ WARNUNG
Tod oder schwere irreversible Verletzungen sind möglich.

⚠ VORSICHT
Leichte reversible Verletzungen sind möglich.

ACHTUNG
Sachschaden ist zu erwarten oder möglich.

!	Wichtige Hinweise auf Fehlfunktionen oder Störungen
i	Weitere Hinweise
► ...	Handlungsaufforderung
> ...	Reaktion, Ergebnis
→ ...	"siehe"
abc	Querverweis
123 0x123 0b010	Dezimalzahl Hexadezimalzahl Binärzahl
[...]	Bezeichnung von Tasten, Schaltflächen oder Anzeigen

1.5 Wie ist diese Dokumentation aufgebaut?

16416
1508

Diese Dokumentation ist eine Kombination aus verschiedenen Anleitungstypen. Sie ist eine Lernanleitung für den Einsteiger, aber gleichzeitig auch eine Nachschlageanleitung für den versierten Anwender. Dieses Dokument richtet sich an die Programmierer der Anwendungen.

Und so finden Sie sich zurecht:

- Um gezielt zu einem bestimmten Thema zu gelangen, benutzen Sie bitte das Inhaltsverzeichnis.
- Mit dem Stichwortregister "Index" gelangen Sie ebenfalls schnell zu einem gesuchten Begriff.
- Am Anfang eines Kapitels geben wir Ihnen eine kurze Übersicht über dessen Inhalt.
- Abkürzungen und Fachbegriffe → Anhang.

Bei Fehlfunktionen oder Unklarheiten setzen Sie sich bitte mit dem Hersteller in Verbindung:
→ www.ifm.com > Land wählen > [Kontakt].

Wir wollen immer besser werden! Jeder eigenständige Abschnitt enthält in der rechten oberen Ecke eine Identifikationsnummer. Wenn Sie uns über Unstimmigkeiten unterrichten wollen, dann nennen Sie uns bitte diese Nummer zusammen mit Titel und Sprache dieser Dokumentation. Vielen Dank für Ihre Unterstützung!

Im Übrigen behalten wir uns Änderungen vor, so dass sich Abweichungen vom Inhalt der vorliegenden Dokumentation ergeben können. Die aktuelle Version finden Sie auf der **ifm**-Homepage:
→ www.ifm.com > Land wählen > [Service] > [Download]
⇨ Unsere Online-Hilfen sind meist "tagesaktuell".
⇨ Die PDF-Handbücher aktualisieren wir nur in großen zeitlichen Abständen.

16420

HINWEIS

Diese Anleitung gilt für das Gerät ohne und mit integriertem E/A-Modul.

► In beiden Fällen die Steuerungskonfiguration unbedingt für das Gerät CR0033 einrichten!

Die Beschreibung zum integrierten E/A-Modul finden Sie hier:
→ Kapitel *Integriertes E/A-Modul: Beschreibung* (→ Seite 240) im Anhang dieser Dokumentation.

1.6 Historie der Anleitung (CR0033 + CR0133)

15794

Was hat sich wann in dieser Anleitung geändert? Ein Überblick:

Datum	Thema	Änderung
2013-06-24	diverse	neue Dokumentenstruktur
2014-02-03	integriertes E/A-Modul	Beschreibung CR0133 hinzugefügt
2014-04-28	diverse FBs	Beschreibung FB-Eingang CHANNEL präzisiert
2014-06-24	FB PID2	Grafik korrigiert
2014-06-30	Name der Dokumentation	"Systemhandbuch" umbenannt zu "Programmierhandbuch"
2014-07-04	Geräte-Ausgang ERROR (Klemme 13)	Ausgang ist nicht vorhanden. Hinweise darauf entfernt.
2014-07-24	FB INPUT_ANALOG FB SET_INPUT_MODE FB SET_OUTPUT_MODE	mögliche Fehler-Codes für Ausgang ERROR ergänzt
2014-07-31	Kapitel "Fehler-Codes, Diagnose"	Fehler-Code-Tabellen ergänzt
2014-07-31	FB PHASE	Beschreibung Parameter der Ausgänge C, ET korrigiert
2014-07-31	FB OUTPUT_CURRENT_CONTROL	Wenn Sollwert=0 mA >> Regelung auf 0 "innerhalb von 100 ms" anstatt "sofort"
2014-08-08	Kapitel "Eingänge integriertes E/A-Modul"	ergänzt um Abschnitte "Analog-Eingänge" und "Binär-Eingänge"
2014-08-08	Kapitel "Objektverzeichnis des integrierten E/A-Moduls"	in den Überschriften "SDOs" ersetzt durch "Objektverzeichnis"
2014-08-26	Beschreibung Eingänge, Ausgänge	highside / lowside ersetzt durch plusschaltend / minusschaltend
2014-08-29	Schnelle Eingänge	Hinweis auf automatisch eingestellten Eingangswiderstand
2014-09-29	Begriff OUT_OVERLOAD_PROTECTION	allgemein ersetzt durch Überlastschutz
2014-11-12	Kapitel "Ausgänge (Technologie)"	Abschnitt "Diagnose der binären Ausgänge" ergänzt oder korrigiert
2015-01-13	Dokumentationsstruktur Fehlercodes, Systemmerker	• Fehlermerker: nur noch im Anhang, Kapitel *Systemmerker* • CAN / CANopen Fehler und Fehlerbehandlung: nur noch im Systemhandbuch "Know-How" • Fehlercodes, EMCY-Codes: nun im Anhang, Kapitel *Fehler-Tabellen*
2015-03-10	Verfügbarer Speicher	Darstellung verbessert
2015-05-26	FB J1939_x_GLOBAL_REQUEST	Beschreibung präzisiert
2015-06-10	diverse FBs	Beschreibung FB-Eingang CHANNEL korrigiert
2015-07-20	Betriebsarten Eingänge IN12...IN15	nun ohne Mode 19
2015-07-27	FB GET_IDENTITY	ergänzt um Ausgang SERIALNUMBER
2015-08-04	Ausgänge OUT00...01	Diagnose via Strom- und Spannungsmessung
2015-08-24	E/A-Modul ExB01	nicht realisierte E/A-Modes entfernt

2 Sicherheitshinweise

Inhalt

213

2.1 Beachten!

214
11212

Mit den in dieser Anleitung gegebenen Informationen, Hinweisen und Beispielen werden keine Eigenschaften zugesichert. Die abgebildeten Zeichnungen, Darstellungen und Beispiele enthalten weder Systemverantwortung noch anwendungsspezifische Besonderheiten.

► Die Sicherheit der Maschine/Anlage muss auf jeden Fall eigenverantwortlich durch den Hersteller der Maschine/Anlage gewährleistet werden.

► Beachten Sie die nationalen Vorschriften des Landes, in welchem die Maschine/Anlage in Verkehr gebracht werden soll!

⚠ WARNUNG

Bei Nichtbeachten der Hinweise in dieser Anleitung sind Sach- oder Körperschäden möglich!
Die **ifm electronic gmbh** übernimmt hierfür keine Haftung.

► Die handelnde Person muss vor allen Arbeiten an und mit diesem Gerät die Sicherheitshinweise und die betreffenden Kapitel dieser Anleitung gelesen und verstanden haben.

► Die handelnde Person muss zu Arbeiten an der Maschine/Anlage autorisiert sein.

► Die handelnde Person muss für die auszuführende Arbeit über die erforderliche Ausbildung und Qualifikation verfügen.

► Beachten Sie die Technischen Daten der betroffenen Geräte!
Das aktuelle Datenblatt finden Sie auf der **ifm**-Homepage:
→ www.ifm.com > Land wählen > [Datenblattsuche] > (Artikel-Nr.) > [Technische Daten im PDF-Format]

► Beachten Sie die Montage- und Anschlussbedingungen sowie die bestimmungsgemäße Verwendung der betroffenen Geräte!
→ mitgelieferte Montageanleitung oder auf der **ifm**-Homepage:
→ www.ifm.com > Land wählen > [Datenblattsuche] > (Artikel-Nr.) > [Betriebsanleitungen]

► Beachten Sie die Korrekturen und Hinweise in den "Release-Notes" zur vorhandenen Hardware, Software und Dokumentation auf der **ifm**-Homepage:
→ www.ifm.com > Land wählen > [Datenblattsuche] > (Artikel-Nr.) > [Betriebsanleitungen]

5020

ACHTUNG

Der Treiberbaustein der seriellen Schnittstelle kann beschädigt werden!

Beim Trennen oder Verbinden der seriellen Schnittstelle unter Spannung kann es zu undefinierten Zuständen kommen, die zu einer Schädigung des Treiberbausteins führen.

► Die serielle Schnittstelle nur im spannungslosen Zustand trennen oder verbinden!

2.2 Welche Vorkenntnisse sind notwendig?

215

Das Dokument richtet sich an Personen, die über Kenntnisse der Steuerungstechnik und SPS-Programmierkenntnisse mit IEC 61131-3 verfügen.

Zum Programmieren der SPS sollten die Personen zusätzlich mit der Software CODESYS vertraut sein.

Das Dokument richtet sich an Fachkräfte. Dabei handelt es sich um Personen, die aufgrund ihrer einschlägigen Ausbildung und ihrer Erfahrung befähigt sind, Risiken zu erkennen und mögliche Gefährdungen zu vermeiden, die der Betrieb oder die Instandhaltung eines Produkts verursachen kann. Das Dokument enthält Angaben zum korrekten Umgang mit dem Produkt.

Lesen Sie dieses Dokument vor dem Einsatz, damit Sie mit Einsatzbedingungen, Installation und Betrieb vertraut werden. Bewahren Sie das Dokument während der gesamten Einsatzdauer des Gerätes auf.

Befolgen Sie die Sicherheitshinweise.

2.3 Anlaufverhalten der Steuerung

15233

11575

⚠ WARNUNG

Gefahr durch unbeabsichtigtes und gefährliches Anlaufen von Maschinen- oder Anlagenteilen!

► Der Programmierer muss bei der Programmerstellung verhindern, dass nach Auftreten eines Fehlers (z.B. NOT-HALT) und der anschließenden Fehlerbeseitigung unbeabsichtigt Maschinen- oder Anlagenteile gefährlich anlaufen können!
⇨ Wiederanlaufsperre realisieren!

► Dazu im Fehlerfall die in Frage kommenden Ausgänge im Programm logisch abschalten!

Ein Wiederanlauf kann z.B. verursacht werden durch:
- Spannungswiederkehr nach Spannungsausfall
- Reset nach Watchdog-Ansprechen wegen zu langer Zykluszeit
- Fehlerbeseitigung nach NOT-HALT

So erreichen Sie sicheres Verhalten der Steuerung:

► Spannungsversorgung im Anwendungsprogramm überwachen.

► Im Fehlerfall alle relevanten Ausgänge im Anwendungsprogramm ausschalten.

► Aktuatoren, die zu gefahrbringenden Bewegungen führen können, zusätzlich im Anwendungsprogramm überwachen (Feedback).

6827

► Relaiskontakte, die zu gefahrbringenden Bewegungen führen können, zusätzlich im Anwendungsprogramm überwachen (Feedback).

► Bei Bedarf im Anwendungsprojekt sicherstellen, dass verschweißte Relaiskontakte keine gefahrbringenden Bewegungen auslösen oder fortführen können.

3 Systembeschreibung

Inhalt

975

3.1 Angaben zum Gerät

10415

Diese Anleitung beschreibt aus der Gerätefamilie für den mobilen Einsatz, **ecomat*mobile*** der **ifm electronic gmbh**:

- ClassicController: CR0033
- ExtendedController: CR0133 (= CR0033 mit integriertem E/A-Modul)

3.2 Hardware-Beschreibung

Inhalt

14081

3.2.1 Hardware-Aufbau

Inhalt

15332

Startvoraussetzung

19658

Das Gerät startet erst, wenn am Versorgungsanschluss VBBS (unter anderem Versorgung der Relais auf der Standardseite) und an Klemme 15 eine ausreichende Spannung anliegt.
Klemme 15 ist in Fahrzeugen die vom Zündschloss geschaltete Plusleitung.

Relais

19661

Der ClassicController verfügt über 2 interne Ausgangsrelais, die jeweils 8 Ausgänge von der Klemmenspannung VBBx trennen können (x = O | R).

Die Relais werden nur unter folgender Voraussetzung aktiviert:
- das globale Bit ERROR = FALSE

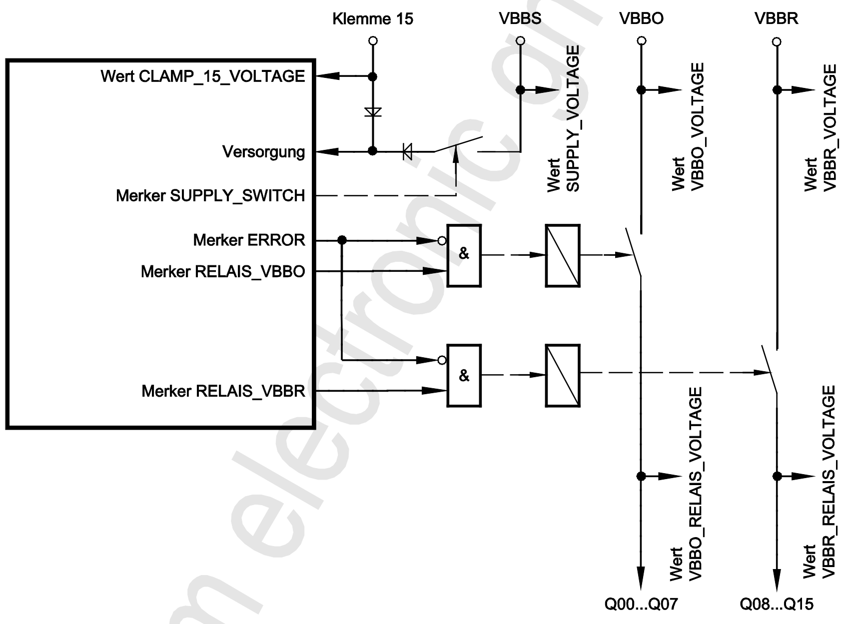
UND
- das Bit RELAIS_VBBx = TRUE

Im aktivierten Zustand legen die Relaiskontakte die Ausgänge an die Klemmenspannung VBBx.

! Zugehörige Ausgänge erst ≥ 45 ms nach Einschalten der Relais aktivieren!

Prinzipschaltung

19662



Grafik: Prinzipaufbau der Versorgung und der Relais

Verfügbarer Speicher

13736

FLASH-Speicher

8136

FLASH-Speicher (nichtflüchtiger, langsamer Speicher) insgesamt im Gerät vorhanden	2 176 kByte

Davon sind folgende Speicherbereiche reserviert für ...

maximale Größe für das Anwendungsprogramm	1 280 kByte
Daten außerhalb des Anwendungsprogramms Anwender kann Daten speichern, z.B. Files, Bitmaps, Fonts	128 kByte
Daten außerhalb des Anwendungsprogramms Daten mit *FLASHREAD* (→ Seite 204) lesen oder mit *FLASHWRITE* (→ Seite 205) schreiben (bei Files: abzüglich 128 Byte für Header)	64 kByte

Der verbleibende Speicher ist reserviert für system-interne Zwecke.

SRAM

8360

SRAM (flüchtiger, schneller Speicher) insgesamt im Gerät vorhanden SRAM steht hier allgemein für alle Arten von flüchtigen, schnellen Speichern.	2 216 kByte

Davon sind folgende Speicherbereiche reserviert für ...

vom Anwendungsprogramm reservierte Daten	192 kByte

Der verbleibende Speicher ist reserviert für system-interne Zwecke.

FRAM

19547

FRAM (nichtflüchtiger, schneller Speicher) insgesamt im Gerät vorhanden FRAM steht hier allgemein für alle Arten von nichtflüchtigen, schnellen Speichern.	128 kByte

Davon sind folgende Speicherbereiche reserviert für ...

im Anwendungsprogramm als VAR_RETAIN deklarierte Variablen	4 kByte
als remanent definierte Merker (ab %MB0...) ▶ Ende des Speicherbereichs im FB *MEMORY_RETAIN_PARAM* (→ Seite 201) angeben!	4 kByte
Vom Anwender frei verfügbarer remanenter Speicher Zugriff erfolgt über *FRAMREAD* (→ Seite 207) und *FRAMWRITE* (→ Seite 208)	16 kByte
Vom Anwender frei verfügbarer remanenter Speicher Zugriff erfolgt über Adressoperator	64 kByte

Der verbleibende Speicher ist reserviert für system-interne Zwecke.

3.2.2 Funktionsweise der verzögerten Abschaltung

993

Werden die Controller von der Versorgungsspannung getrennt, werden im Normalfall sofort alle Ausgänge abgeschaltet, keine Eingangssignale mehr eingelesen und die Abarbeitung der Steuerungssoftware (Laufzeitsystem und Anwendungsprogramm) abgebrochen. Dieses geschieht unabhängig davon, in welchem Programmschritt sich der Controller befindet.

Wenn dieses Verhalten nicht gewünscht ist, muss der Controller programmgesteuert abgeschaltet werden. Das ermöglicht nach Abschalten der Zündung zum Beispiel das Sichern von Speicherständen.

Die ClassicController können durch eine entsprechende Beschaltung der Versorgungsspannungs-Eingänge und die Auswertung der zugehörigen Systemmerker, programmgesteuert abgeschaltet werden. Das Prinzipschaltbild im Kapitel *Hardware-Aufbau* (→ Seite 13) zeigt schematisch die Zusammenhänge der einzelnen Strompfade.

Klemme VBB15 mit Zündschalter verbinden

2418

Über die Klemme VBB15 wird die interne Steuerungselektronik initialisiert, wenn an Klemme VBBS Versorgungsspannung anliegt.

Diese Klemmen VBB15 und VBBS werden intern überwacht. Die anliegende Klemmenspannung VBB15 kann über den Systemmerker CLAMP_15_VOLTAGE überwacht werden. Die anliegende Klemmenspannung VBBS kann über den Systemmerker SUPPLY_VOLTAGE überwacht werden.

Selbsthaltung

2419

Einschalten der Steuerung:

- Der Zündschalter legt Spannung an VBB15 (Klemme 15*).
- Der Systemmerker CLAMP_15_VOLTAGE erkennt die angelegte Spannung und aktiviert den Systemmerker SUPPLY_SWITCH.
- SUPPLY_SWITCH aktiviert die Verbindung zum Potential VBBS.

> Somit ist der Zündschalter überbrückt, die Selbsthaltung der Steuerspannung ist hergestellt.

Ausschalten der Steuerung über Klemme 15:

- Der Systemmerker CLAMP_15_VOLTAGE erkennt das Abschalten der Versorgungsspannung an Klemme VBB15.

► Im Anwendungsprogramm den Systemmerker SUPPLY_SWITCH zurücksetzen.

> Die Selbsthaltung über VBBS ist aufgehoben und der Controller wird vollständig abgeschaltet.

*) Klemme 15 ist in Fahrzeugen die vom Zündschloss geschaltete Plusleitung.

3.2.3 Relais: wichtige Hinweise!

12976

Zuordnung Relais – Potentiale: → Datenblatt
Max. Summenstrom je Relaiskontakt (= je Ausgangsgruppe): → Datenblatt

ACHTUNG

Gefahr der Zerstörung der Relaiskontakte!
"Klebende" Relaiskontakte können auch im Notfall nicht mehr die Ausgänge von der Versorgung trennen!

Falls VBBS (VBBrel) und Klemme 15 gleichzeitig von der Versorgung getrennt werden, jedoch die Potentiale VBBx an der Versorgung angeschlossen bleiben, dann können die Relais schon abfallen, bevor die Ausgänge vom System deaktiviert werden.

In diesem Fall trennen die Relais **unter Last** die Ausgänge von der Versorgung. Dies schränkt die Lebensdauer der Relais deutlich ein.

► Bei dauerhaftem Anschluss von VBBx an Versorgung:
  • auch VBBS (VBBrel) dauerhaft anschließen und
  • die Ausgänge programmgesteuert mit Hilfe von Klemme 15 abschalten.

3.2.4 Überwachungskonzept

Inhalt

991

Die Steuerung überwacht die Versorgungsspannungen und die System-Fehlermerker. Je nach Zustand ...

- die Steuerung schaltet die internen Relais ab
  - > die Ausgänge werden stromlos, behalten aber ihren logischen Zustand
  - > das Programm läuft weiter

oder:

- die Steuerung schaltet vollständig ab
  - > das Programm stoppt
  - > die Ausgänge werden stromlos und gehen auf logisch "0"
  - > die Status-LED erlischt

Überwachung der Versorgungsspannungen

6752

Wir unterscheiden 2 Szenarien:

Klemmenspannung VBBx fällt unter den Grenzwert von 5,25 V

15752

> Die Steuerung erkennt Unterspannung. Die von der Klemmenspannung VBBx versorgten Ausgänge werden deaktiviert.

> Erholt sich die Klemmenspannung und befindet sich wieder im regulären Bereich (> 10 V), werden die Ausgänge wieder aktiviert.

13975

⚠ **WARNUNG**

Gefährlicher Wiederanlauf möglich!
Gefahr von Personenschaden! Gefahr von Sachschaden an der Maschine/Anlage!

Wird ein Ausgang im Fehlerfall hardwaremäßig abgeschaltet, ändert sich der durch das Anwendungsprogramm erzeugte logische Zustand dadurch nicht.

► Abhilfe:
- Die Ausgänge zunächst im Anwendungsprogramm logisch zurücksetzen!
- Fehler beseitigen!
- Ausgänge situationsabhängig wieder setzen.

Versorgungsspannung VBBS fällt unter den Grenzwert von 10 V

20638

> Die Steuerung läuft weiter, bis die Spannung so weit gefallen ist, dass die daraus erzeugten internen Spannungen einbrechen.

! Unterhalb von 10 V werden keine Retain-Daten gespeichert. → Merker RETAIN_WARNING

> Brechen die internen Spannungen ein, geht der Controller in den Reset.
> Die Ausführung von Laufzeitsystem und Anwendungsprogramm wird abgebrochen.
> Dies geschieht unabhängig davon, in welchem Programmschritt sich die Steuerung befindet.

> Ein Wiederanlauf der Steuerung erfolgt erst, wenn die Versorgungsspannungen wieder oberhalb des Grenzwerts sind.

Überwachungs- und Sicherungsmechanismen

2421

⚠ WARNUNG

Gefahr durch unbeabsichtigtes Abschalten aller Ausgänge!

Falls Überwachungsroutinen einen Systemfehler feststellen:

> das Gerät schaltet die Energie für alle Ausgänge aus.

Während des Programmablaufes stehen die Ausgangsrelais unter voller Software-Kontrolle des Anwenders. So kann z.B. ein paralleler Kontakt der Sicherheitskette als Eingangssignal ausgewertet und das Ausgangsrelais entsprechend abgeschaltet werden. Zur weiteren Sicherheit müssen die entsprechenden nationalen Vorschriften beachtet werden.

Tritt während des Programmablaufs ein Fehler auf, können durch das Systemmerker-Bit ERROR die Relais spannungsfrei geschaltet werden, um kritische Anlagenteile abzutrennen.

Manuelles Setzen von einem Merker-Bit ERROR_VBB... hat KEINE Auswirkungen auf die Relais!

11575

⚠ WARNUNG

Gefahr durch unbeabsichtigtes und gefährliches Anlaufen von Maschinen- oder Anlagenteilen!

- ► Der Programmierer muss bei der Programmerstellung verhindern, dass nach Auftreten eines Fehlers (z.B. NOT-HALT) und der anschließenden Fehlerbeseitigung unbeabsichtigt Maschinen- oder Anlagenteile gefährlich anlaufen können!
  ⇨ Wiederanlaufsperre realisieren!
- ► Dazu im Fehlerfall die in Frage kommenden Ausgänge im Programm logisch abschalten!

Bei Auftreten eines Watchdog-Fehlers ...

> die Programmabarbeitung wird automatisch unterbrochen
> die Ausgänge werden stromlos und gehen auf logisch "0"
> der Controller wird zurückgesetzt
> der Controller startet anschließend neu, wie nach einem Power-On.

Referenzspannungsausgang

2250
13934

Der Referenzspannungsausgang dient der Versorgung von Sensoren mit einer stabilen Spannung, die nicht den Schwankungen der Versorgungsspannung unterworfen ist.

13402

ACHTUNG

Referenzspannungsausgang kann beschädigt werden!

► Von außen KEINE Spannung anlegen!

Über die binären Systemvariablen REFERENCE_VOLTAGE_5 oder REFERENCE_VOLTAGE_10 wird die Spannung am Referenzspannungsausgang [V_{REF} OUT] eingestellt:

REFERENZ_VOLTAGE_10	REFERENZ_VOLTAGE_5	Referenzspannung [V_{REF} OUT]
FALSE	FALSE	0 V
FALSE	TRUE	5 V
TRUE	FALSE	10 V
TRUE	TRUE	0 V

► Wenn Referenzspannungsausgang = 10 V gewählt:
die Steuerung mit mindestens 13 V versorgen!

► Überwachen der Spannung am Referenzspannungsausgang mit Systemvariable REF_VOLTAGE.

> Wenn Systemvariable ERROR = TRUE:
der Referenzspannungsausgang wird deaktiviert (Ausgabe = 0 V).

3.2.5 Eingänge (Technologie)

14090

Analog-Eingänge

15446

Die Analog-Eingänge können über das Anwendungsprogramm konfiguriert werden. Der Messbereich kann zwischen folgenden Bereichen umgeschaltet werden:

- Stromeingang 0...20 mA
- Spannungseingang 0...10 V
- Spannungseingang 0...32 V
- Widerstandsmessung 16...30 000 Ω / 3...680 Ω (Messung gegen GND)

Die Spannungsmessung kann auch ratiometrisch erfolgen (0...1000 ‰, über FBs einstellbar). Das bedeutet, ohne zusätzliche Referenzspannung können Potentiometer oder Joysticks ausgewertet werden. Ein Schwanken der Versorgungsspannung hat auf diesen Messwert keinen Einfluss.

Alternativ kann ein Analog-Kanal auch binär ausgewertet werden.

> ! Bei ratiometrischer Messung müssen die angeschlossenen Sensoren mit VBBS des Geräts versorgt werden. Dadurch werden Fehlmessungen durch Spannungsverschiebungen vermieden.

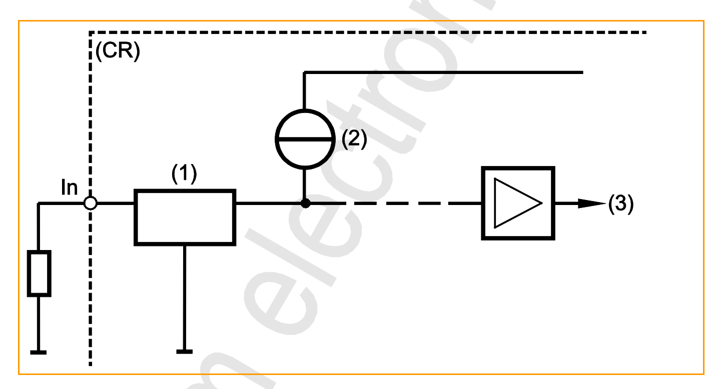
8971

(CR)
(+)
(6)
(5)
In
(1)
(2)
(3a)
(3b)
(4a)
(4b)

In = Anschluss Multifunktions-Eingang n
(CR) = Gerät
(1) = Eingangsfilter
(2) = analoge Strommessung
(3a) = Binär-Eingang plus-schaltend
(3b) = Binär-Eingang minus-schaltend
(4a) = analoge Spannungsmessung 0...10 V
(4b) = analoge Spannungsmessung 0...32 V
(5) = Spannung
(6) = Referenz-Spannung

Grafik: Prinzipschaltung Multifunktions-Eingang

8972



Grafik: Prinzipschaltung Widerstandsmess-Eingang

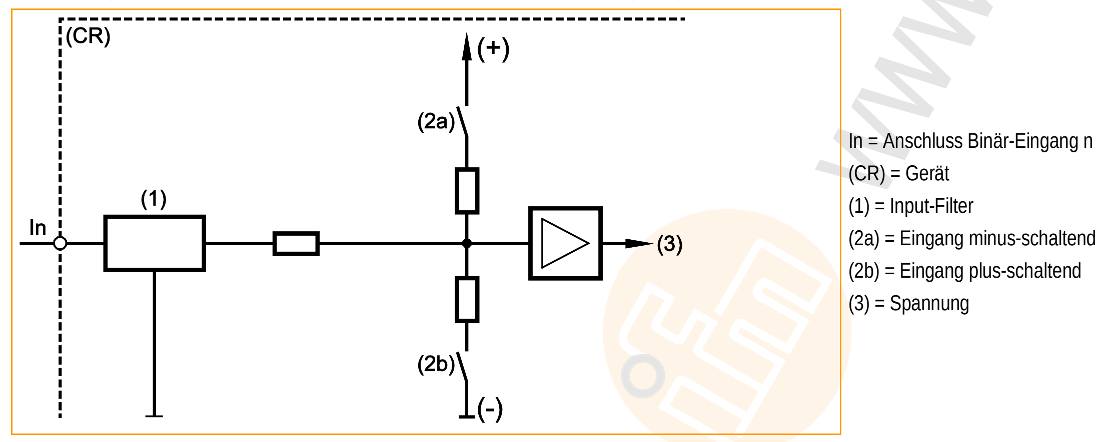
In = Anschluss Widerstandsmess-Eingang n
(CR) = Gerät
(1) = Eingangsfilter
(2) = Konstantstromquelle
(3) = Spannung

Binär-Eingänge

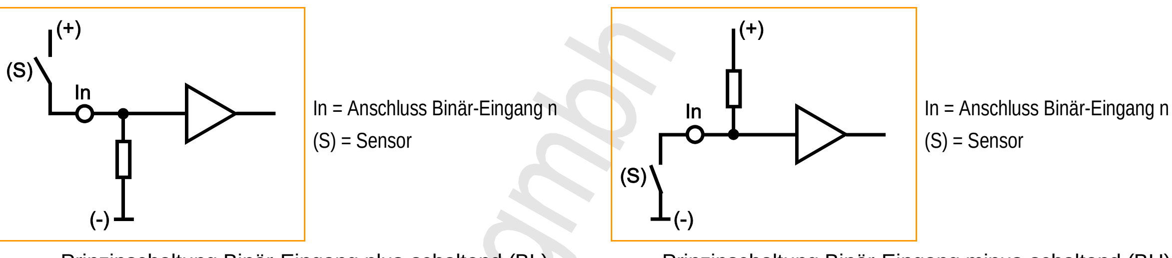
1015
7345

Der Binär-Eingang kann in folgenden Modi betrieben werden:
- binärer Eingang plus-schaltend (BL) für positives Gebersignal
- binärer Eingang, minus-schaltend (BH) für negatives Gebersignal

Je nach Gerät können auch die Binär-Eingänge unterschiedlich konfiguriert werden. Neben den Schutzmechanismen gegen Störungen werden die Binär-Eingänge intern über eine Analogstufe ausgewertet. Das ermöglicht die Diagnose der Eingangssignale. Im Anwendungsprogramm steht das Schaltsignal aber direkt als Bit-Information zur Verfügung.



Grafik: Prinzipschaltung Binär-Eingang minus-schaltend / plus-schaltend für negative und positive Gebersignale



Prinzipschaltung Binär-Eingang plus-schaltend (BL)
für positives Sensorsignal:
Eingang = offen ⇨ Signal = Low (Supply)

Prinzipschaltung Binär-Eingang minus-schaltend (BH)
für negatives Sensorsignal:
Eingang = offen ⇨ Signal = High (GND)

Bei einem Teil dieser Eingänge (→ Datenblatt) kann das Potential gewählt werden, gegen das geschaltet wird.

Eingangsgruppe I0 (I00...11)

10423

Bei diesen Eingängen handelt es sich um eine Gruppe von Multifunktionskanälen.

Jeder einzelne dieser Eingänge ist wahlweise wie folgt konfigurierbar:
- analoger Eingang 0...20 mA
- analoger Eingang 0...10 V
- analoger Eingang 0...32 V
- Spannungsmessung ratiometrisch 0...1000 ‰
- binärer Eingang plus-schaltend (BL) für positives Gebersignal (mit/ohne Diagnose)
- binärer Eingang, minus-schaltend (BH) für negatives Gebersignal
- schneller Eingang für z.B. Inkrementalgeber und Frequenz- oder Periodendauermessung

→ Kapitel *Mögliche Betriebsarten Ein-/Ausgänge* (→ Seite 232)

Diagnosefähige Sensoren nach NAMUR können ausgewertet werden.

Alle Eingänge zeigen das gleiche Verhalten bei Funktion und Diagnose. Jedoch:
I00...I07: Multifunktionseingang mit versorgungsspannungsabhängigen Pegeln zur Frequenzmessung.
I08...I11: Multifunktionseingang mit festen Pegeln zur Frequenzmessung.

ℹ Detaillierte Beschreibung → Kapitel *Adressbelegung Ein-/Ausgänge* (→ Seite 229)

► Die Konfiguration jedes einzelnen Eingangs erfolgt über das Anwendungsprogramm:
- FB *INPUT_ANALOG* (→ Seite 129) > Eingang MODE oder:
- FB *SET_INPUT_MODE* (→ Seite 132) > Eingang MODE
- schnelle Eingänge mit folgenden FBs:

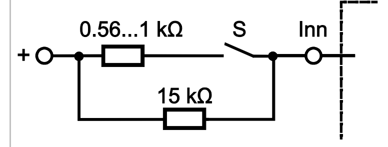
FAST_COUNT (→ Seite 140)	Zählerbaustein für schnelle Eingangsimpulse
FREQUENCY (→ Seite 142)	misst die Frequenz des am gewählten Kanal ankommenden Signals
FREQUENCY_PERIOD (→ Seite 144)	misst die Frequenz und die Periodendauer (Zykluszeit) in [µs] am angegebenen Kanal
INC_ENCODER (→ Seite 146)	Vorwärts-/Rückwärts-Zählerfunktion zur Auswertung von Drehgebern
PERIOD (→ Seite 148)	misst am angegebenen Kanal die Frequenz und die Periodendauer (Zykluszeit) in [µs]
PHASE (→ Seite 152)	liest ein Kanalpaar mit schnellen Eingängen ein und vergleicht die Phasenlage der Signale

► Für NAMUR: Soll die Diagnose genutzt werden, dann diesen Modus zusätzlich aktivieren:
- FB SET_INPUT_MODE > Eingang DIAGNOSTICS setzen.

> Werden die Analogeingänge auf Strommessung konfiguriert, wird bei Überschreiten des Endwertes (21,7 mA) in den sicheren Spannungsmessbereich (0...32 V DC) geschaltet und das jeweilige Fehlerbit im Merkerbyte ERROR_CURRENT_Ix gesetzt. Sinkt der Wert wieder unter den Grenzwert, schaltet der Eingang selbsttätig auf den Strommessbereich zurück.

NAMUR-Diagnose für digitale Signale bei nichtelektronischen Schaltern:

► Schalter mit einer zusätzlichen Widerstandsbeschaltung versehen!



Grafik: Nichtelektronischer Schalter S am Eingang Inn

13956

> Das Diagnose-Ergebnis zeigen z.B. folgende Systemmerker:

Systemmerker (Symbolname)	Typ	Beschreibung
ERROR_BREAK_Ix (x=0...n; Wert abhängig vom Gerät, → Datenblatt)	DWORD	Eingangsgruppe x: Leiterbruch-Fehler oder (Widerstandseingang): Schluss nach Versorgung [Bit 0 für Eingang 0] ... [Bit z für Eingang z] dieser Gruppe Bit = TRUE: Fehler Bit = FALSE: kein Fehler
ERROR_SHORT_Ix (x=0...n; Wert abhängig vom Gerät, → Datenblatt)	DWORD	Eingangsgruppe x: Kurzschluss-Fehler [Bit 0 für Eingang 0] ... [Bit z für Eingang z] dieser Gruppe Bit = TRUE: Fehler Bit = FALSE: kein Fehler

> Im Anwendungsprogramm können die Systemvariablen ANALOG00...ANALOGxx zur kundenspezifischen Diagnose der Eingänge dienen.

Eingangsgruppe I1 (I12...15)

10424

Bei diesen Eingängen handelt es sich um eine Gruppe von Multifunktionskanälen.

Jeder einzelne dieser Eingänge ist wahlweise wie folgt konfigurierbar:
- binärer Eingang plus-schaltend (BL) für positives Gebersignal (mit/ohne Diagnose)
- Eingang für Widerstandsmessung (z.B. Temperatursensoren oder Tankgeber)

→ Kapitel *Mögliche Betriebsarten Ein-/Ausgänge* (→ Seite 232)

Diagnosefähige Sensoren nach NAMUR können ausgewertet werden.

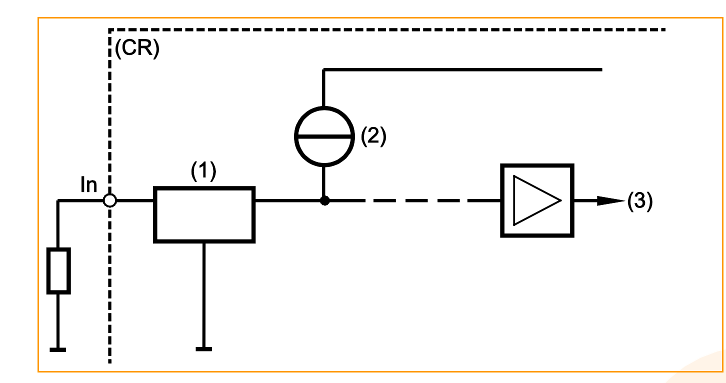
► Die Konfiguration jedes einzelnen Eingangs erfolgt über das Anwendungsprogramm:
  - FB *INPUT_ANALOG* (→ Seite 129) > Eingang MODE oder:
  - FB *SET_INPUT_MODE* (→ Seite 132) > Eingang MODE

Widerstandsmessung

9773

Typische Sensoren an diesen Eingängen:
- Tankpegel
- Temperatur (PT1000, NTC)

8972

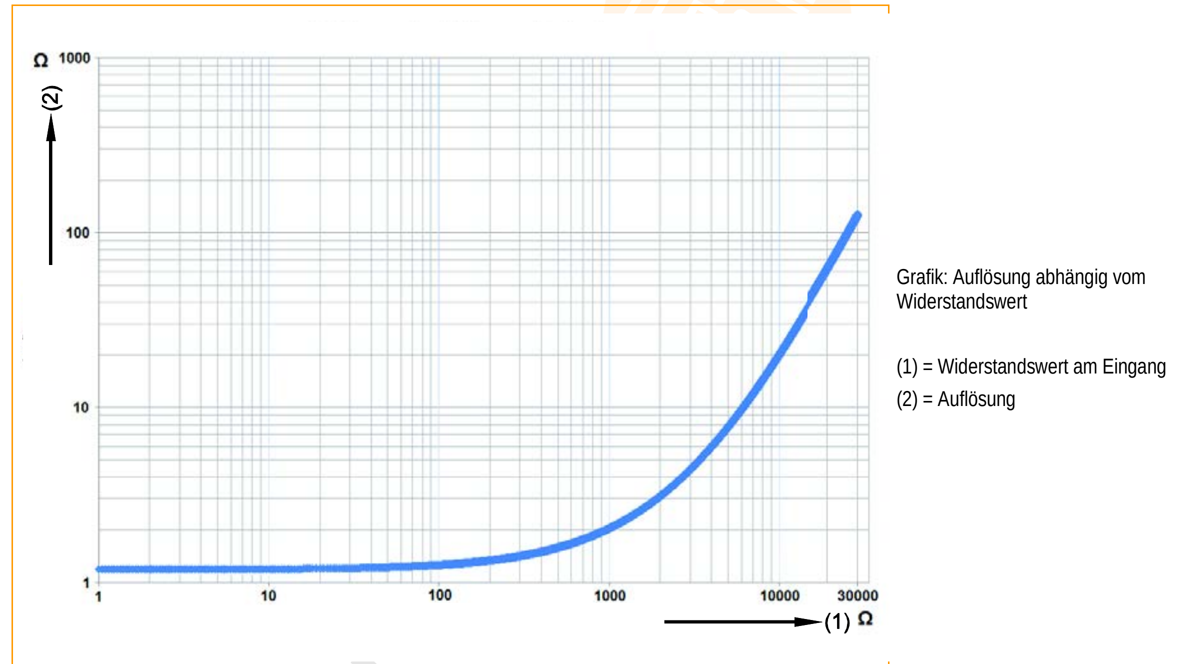


Grafik: Prinzipschaltung Widerstandsmess-Eingang

In = Anschluss Widerstandsmess-Eingang n
(CR) = Gerät
(1) = Eingangsfilter
(2) = Konstantstromquelle
(3) = Spannung

8970

Bei diesem Gerät ist die Auflösung nicht linear abhängig vom Widerstandswert, → Grafik:



Grafik: Auflösung abhängig vom Widerstandswert

(1) = Widerstandswert am Eingang
(2) = Auflösung

Um wieviel Ohm ändert sich der Messwert, wenn sich das Signal des A/D-Wandlers am Eingang um 1 ändert? Beispiele:
- Im Bereich 1...100 Ω beträgt die Auflösung 1,2 Ω.
- Im Bereich bei 1 kΩ beträgt die Auflösung ca. 2 Ω.
- Im Bereich bei 2 kΩ beträgt die Auflösung ca. 3 Ω.
- Im Bereich bei 3 kΩ beträgt die Auflösung ca. 6 Ω.
- Im Bereich bei 6 kΩ beträgt die Auflösung ca. 10 Ω.
- Im Bereich bei 10 kΩ beträgt die Auflösung ca. 11 Ω
- Im Bereich bei 20 kΩ beträgt die Auflösung ca. 60 Ω.

3.2.6 Ausgänge (Technologie)

Inhalt

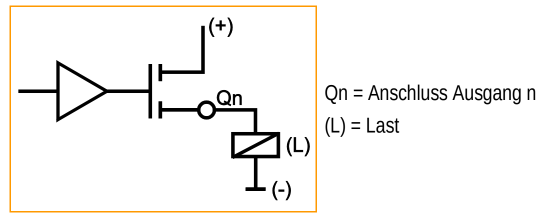
14093

Binär-Ausgänge

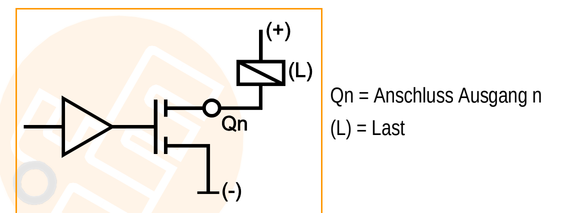
14094

Bei den Geräte-Ausgängen sind folgende Betriebsarten möglich (→ Datenblatt):
- binärer Ausgang, plus-schaltend (BH) mit/ohne Diagnosefunktion
- binärer Ausgang, minus-schaltend (BL) ohne Diagnosefunktion

15450



Prinzipschaltung Ausgang plus-schaltend (BH)
für positives Ausgangssignal



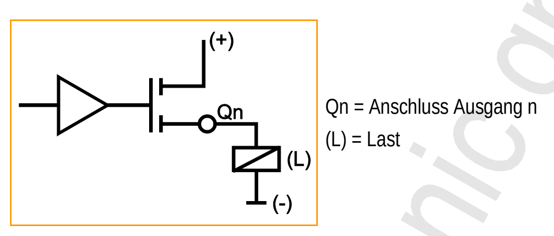
Prinzipschaltung Ausgang minus-schaltend (BL)
für negatives Ausgangssignal

PWM-Ausgänge

14095

Bei den Geräte-Ausgängen sind folgende Betriebsarten möglich (→ Datenblatt):
- PWM-Ausgang, plus-schaltend (BH) ohne Diagnosefunktion

15451



Prinzipschaltung Ausgang plus-schaltend (BH)
für positives Ausgangssignal

Schutzfunktionen der Ausgänge

15248

Die Ausgänge dieses Geräts sind in Grenzen gegen Überlast und Kurzschluss geschützt.
→ Datenblatt

Definition: Überlast

15249

Überlast kann nur an einem Ausgang mit Strommessung erkannt werden.

Überlast ist definiert als ...
"nominaler Maximalstrom laut Datenblatt + 12,5 %".

Definition: Kurzschluss

15250

Ein Kurzschluss kann an allen diagnosefähigen Ausgängen erkannt werden und ist wie folgt definiert:

Kurzschluss ist definiert als ...
"Absinken der Ausgangsspannung unter 88 % (± 2,5 % vom gemessenen Wert) der zugehörigen Versorgungsspannung."

> Ein Schluss gegen Masse kann nur erkannt werden bei Ausgang = TRUE.

Reaktion der Ausgänge auf Überlast oder Kurzschluss

15251

Eigenschutz des Ausgangs

15333

Unabhängig von der Betriebsart des Ausgangs und der Fehlererkennung schützt sich die Hardware selbst. Bei zu hoher thermischer Belastung (durch Kurzschluss oder Überlast) beginnt der Ausgangstreiber zu takten.
(!) Bei zu lange andauerndem Takten des Ausgangs (mehrere Stunden) kann der Treiber beschädigt werden!

Wir empfehlen deshalb:
Diagnosefähige Ausgänge des Geräts unbedingt mit folgenden Einstellungen betreiben, da hier die Software zusätzlich die Treiber durch Abschalten schützt:
- FB *SET_OUTPUT_MODE* (→ Seite 155) > Eingang DIAGNOSTICS = TRUE und
- FB SET_OUTPUT_MODE > Eingang PROTECTION = TRUE

Reaktion abhängig von Betriebsart des Ausgangs

15479

Im Falle von Überlast oder Kurzschluss hängt das Verhalten des Ausgangs von dessen Betriebsart ab (→ FB *SET_OUTPUT_MODE* (→ Seite 155) > Eingänge DIAGNOSTICS und PROTECTION):

- DIAGNOSTICS = FALSE und PROTECTION = FALSE:
  > der Ausgang wird weiter betrieben.
- DIAGNOSTICS = TRUE und PROTECTION = FALSE:
  > Fehler wird erkannt und als Fehler-Code gemeldet (→ Kapitel *Fehler-Codes* (→ Seite 306)).
  Das hängt vom Ausgangstyp und dem Strom oder der Spannung am Ausgang ab.
  Der Programmierer kann im Programm auf den Fehler reagieren.
- DIAGNOSTICS = TRUE und PROTECTION = TRUE:
  > Fehler wird erkannt und als Fehler-Code gemeldet (→ Kapitel Fehler-Codes).
  > Der betreffende Ausgang wird abgeschaltet.
  > (!) Der logische Zustand des Ausgangs bleibt davon unverändert!
  > Die Steuerung prüft im Sekundentakt, ob der Fehler beseitigt ist.
  Falls Fehler beseitigt: Steuerung gibt Ausgang wieder frei.

Reaktion bei Einsatz von PWM1000

15480

Anders verhält es sich bei Einsatz des FBs *PWM1000* (→ Seite 167):
Hier gibt es keine Diagnose.
Der *Eigenschutz des Ausgangs* (→ Seite 27) wird aktiv.

Reaktion bei Ausgängen mit Stromrücklesung

20641

► Bei Ausgängen mit Stromrücklesung:
Im Anwendungsprogramm den typischen Strom für den Ausgang abfragen!
Hier ist der Anwendungsprogrammierer verantwortlich, auf das Ereignis zu reagieren.

Ausgangsgruppe Q0 (Q00...15)

10445
12295

Bei diesen Ausgängen handelt es sich um eine Gruppe von Multifunktionskanälen.

Jeder einzelne dieser Ausgänge ist wahlweise wie folgt konfigurierbar:
- binärer Ausgang, plus-schaltend (BH), teilweise auch minus-schaltend (BL)
- analoger Ausgang, stromgeregelt (PWMi)
- analoger Ausgang mit Pulsweitenmodulation (PWM), teilweise als H-Brücke

→ Kapitel *Mögliche Betriebsarten Ein-/Ausgänge* (→ Seite 232)

► Die Konfiguration jedes einzelnen Ausgangs erfolgt über das Anwendungsprogramm:
→ FB *SET_OUTPUT_MODE* (→ Seite 155) > Eingang MODE
Lastströme anzeigen → FB *OUTPUT_CURRENT* (→ Seite 163)
PWM-Ausgang: → FB *PWM1000* (→ Seite 167)
PWMi-Ausgang: → FB *OUTPUT_CURRENT_CONTROL* (→ Seite 164)
H-Brücke steuern → FB *OUTPUT_BRIDGE* (→ Seite 159)

► Strommessbereich konfigurieren für die Ausgänge Q00…Q03 und Q08…Q11 (wahlweise 2 A oder 4 A):
→ FB SET_OUTPUT_MODE > Eingang CURRENT_RANGE

Bei Einsatz der H-Brücke wird die Stromregelung nicht unterstützt.

HINWEIS

Um die internen Messwiderstände zu schützen, sollte der Überlastschutz immer aktiv sein (voreingestellt). Je nach gewähltem Strommessbereich besteht Schutz ab 2,25 A oder ab 4,5 A.
Die Funktion wird **nicht** im PWM-Modus unterstützt.
Die Funktion kann bei Bedarf abgeschaltet werden.

Zu den Grenzwerten unbedingt das Datenblatt beachten!

Die Leiterbruch- und die Kurzschlusserkennung sind aktiv, wenn...
- der Ausgang ist als "binär plusschaltend" (BH) konfiguriert UND
- der Ausgang ist EINgeschaltet.

► Bei Verwendung von gegen Masse schaltenden Ausgängen darf die Versorgungsspannung an der angeschlossenen Last nicht höher sein als die Versorgungsspannung(en) der Ausgangsgruppe(n)!

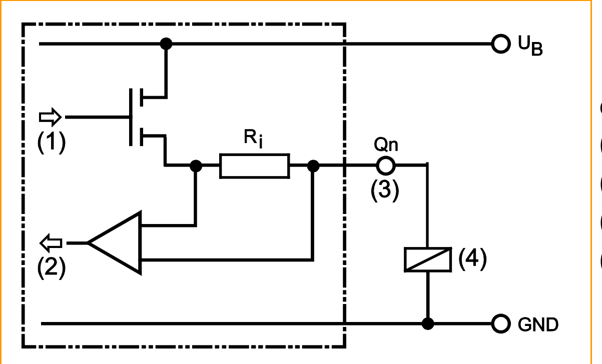
13976

Abhängig von der Umgebungstemperatur kann ab einem bestimmten Kurzschlussstrom ein Kurzschluss eventuell nicht mehr zuverlässig erkannt werden, da die Ausgangstreiber sich zum Schutz vor Zerstörung selbsttätig zeitweise deaktivieren.

Diagnose: binäre Ausgänge (via Strommessung)

19398
19396

Die Diagnose dieser Ausgänge erfolgt über eine interne Strommessung im Ausgang:



Grafik: Prinzipschaltung
(1) Ausgangskanal
(2) Rücklesekanal für Diagnose
(3) Anschluss Ausgang
(4) Last

Diagnose: Überlast (via Strommessung)

19437
15249

Überlast kann nur an einem Ausgang mit Strommessung erkannt werden.

Überlast ist definiert als ...
"nominaler Maximalstrom laut Datenblatt + 12,5 %".

Diagnose: Leiterbruch (via Strommessung)

19400

Eine Leiterbruch-Erkennung erfolgt über den Rücklesekanal. Bei geschaltetem Ausgang (Qn=TRUE) wird dann ein Leiterbruch erkannt, wenn über den Widerstand R_i kein Strom fließt (keine Spannung abfällt). Ohne den Leiterbruch fließt durch den Längswiderstand R_i der Laststrom und erzeugt damit einen Spannungsabfall, der über den Rücklesekanal ausgewertet wird.

Diagnose: Kurzschluss (via Strommessung)

19401

Eine Kurzschluss-Erkennung erfolgt über den Rücklesekanal. Bei geschaltetem Ausgang (Qn=TRUE) wird dann ein Kurzschluss gegen GND erkannt, wenn über den Längswiderstand R_i die Versorgungsspannung abfällt.

3.2.7 Hinweise zur Anschlussbelegung

1426

Die Anschlussbelegungen (→ Montageanleitungen der Geräte, Kapitel "Anschlussbelegung") beschreiben die Standard-Gerätekonfigurationen. Die Anschlussbelegung dient der Zuordnung der Ein- und Ausgangskanäle zu den IEC-Adressen und den Geräteanschlussklemmen.

Die einzelnen Kürzel haben folgende Bedeutung:

A	Analog-Eingang
BH	Binärer highside-Eingang: minus-schaltend für negatives Sensorsignal Binärer highside-Ausgang: plus-schaltend für positives Ausgangssignal
BL	Binärer lowside-Eingang: plus-schaltend für positives Sensorsignal Binärer lowside-Ausgang: minus-schaltend für negatives Ausgangssignal
CYL	Eingang Periodendauermessung
ENC	Eingang Drehgebersignale
FRQ	Frequenzeingang
H-Bridge	Ausgang mit H-Brücken-Funktion
PWM	**P**uls**w**eiten-**m**oduliertes Signal
PWMi	PWM-Ausgang mit Strommessung
IH	Impuls-/Zählereingang, highside, minus-schaltend für negatives Sensorsignal
IL	Impuls-/Zählereingang, lowside, plus-schaltend für positives Sensorsignal
R	Rücklesekanal für einen Ausgang

Zuordnung der Ein-/Ausgangskanäle: → Katalog, Montageanleitung oder Datenblatt

3.2.8 Sicherheitshinweise zu Reed-Relais

7348

Beim Einsatz von nichtelektronischen Schaltern Folgendes beachten:

> ! Kontakte von Reed-Relais können (reversibel) verkleben, wenn sie ohne Vorwiderstand an den Geräte-Eingängen angeschlossen werden.

► **Abhilfe:** Vorwiderstand zum Reed-Relais installieren:
Vorwiderstand = max. Eingangsspannung / zulässiger Strom im Reed-Relais
Beispiel: 32 V / 500 mA = 64 Ohm

► Der Vorwiderstand darf 5 % des Eingangswiderstands RE des Geräte-Eingangs (→ Datenblatt) nicht überschreiten. Sonst wird das Signal nicht als TRUE erkannt.
Beispiel:
RE = 3 000 Ohm
⇨ max. Vorwiderstand = 150 Ohm

3.2.9 Rückspeisung bei extern beschalteten Ausgängen

2422

In manchen Anwendungen werden Aktuatoren nicht nur von Ausgängen der SPS gesteuert, sondern zusätzlich von externen Schaltern. In solchen Fällen müssen die extern beschalteten Ausgänge mit Sperrdioden geschützt werden (→ Grafik unten).

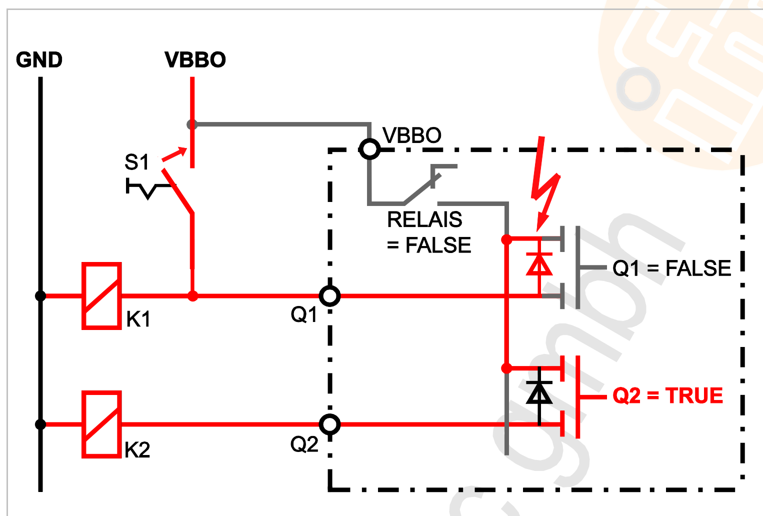
ACHTUNG

Zerstörung von Ausgängen bei unzulässiger Rückspeisung!

Werden Aktoren von extern angesteuert, darf die Potentialschiene derselben Ausgangsgruppe nicht potentialfrei werden (z.B. bei RELAIS = FALSE).

Andernfalls findet über die integrierte Schutzdiode im Ausgangstreiber des extern beschalteten Ausgangs eine Rückspeisung der Klemmenspannung VBB auf die Potentialschiene der Ausgangsgruppe statt. Dadurch steuert ein gesetzter anderer Ausgang derselben Gruppe seine an ihm angeschlossene Last an. Durch den Laststrom wird der rückspeisende Ausgang zerstört.

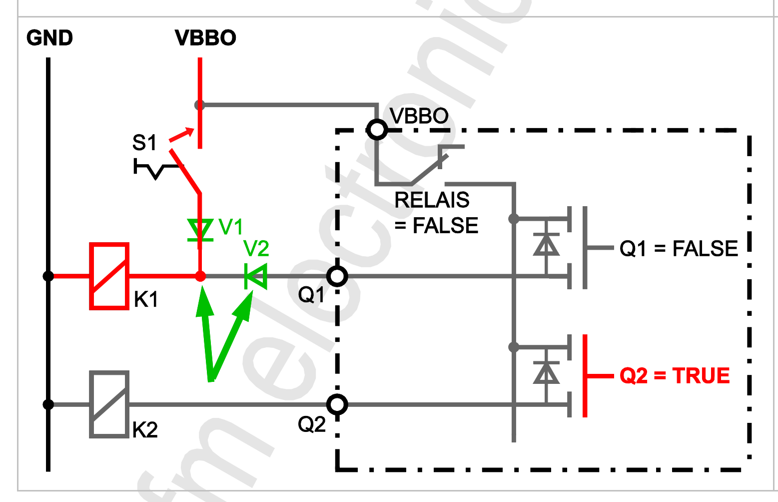
► Extern beschaltete Ausgänge mit Sperrdioden schützen!



Beispiel:
Merker RELAIS schaltet die Versorgung VBBO der Ausgangsgruppe aus.

Ohne Sperrdioden speist der externe Schalter S1 die Versorgung VBBO über die interne Schutzdiode (rot) von Ausgang Q1 auf die interne Potentialschiene der Ausgänge.

Wird Ausgang Q2 = TRUE (→ Grafik), dann bekommt K2 trotz RELAIS = FALSE Spannung über die Schutzdiode von Q1 (rote Linien). Wegen Überlastung brennt diese Schutzdiode durch und der Ausgang Q1 wird zerstört!



Grafik: Beispiel Beschaltung mit Sperrdioden wegen Gefahr der Rückspeisung

Abhilfe:
Sperrdioden V1 und V2 einsetzen (→ grüne Pfeile)!

Erfolg:
Wenn RELAIS = FALSE, dann bleibt K2 ausgeschaltet, auch wenn Q2 = TRUE.

HINWEIS

Abhilfe bei extern beschalteten Ausgängen

► Die extern beschalteten Ausgänge so über Dioden entkoppeln, dass keine externe Spannung an die Ausgangsklemme der Steuerung geschaltet werden kann!

3.2.10 Status-LED

1430

Die Betriebszustände werden durch die integrierte Status-LED (Default-Einstellung) angezeigt.

LED-Farbe	Blinkfrequenz	Beschreibung
aus	konstant aus	keine Betriebsspannung
Gelb	kurzzeitig ein	Initialisierung oder Reset Checks
Grün / schwarz	5 Hz	kein Laufzeitsystem geladen
Grün / schwarz	2 Hz	Anwendung RUN
Grün	konstant ein	Anwendung STOP
Rot / schwarz	2 Hz	Anwendung RUN mit Fehler
Rot	kurzzeitig ein	Fatal Error
Rot	konstant ein	Fatal Error (bei TEST-Eingang aktiv) ERROR STOP / SYSTEM STOP

Die Betriebszustände STOP und RUN können vom Programmiersystem geändert werden.

LED im Anwendungsprogramm steuern

13142

Bei diesem Gerät kann die Status-LED auch durch das Anwendungsprogramm gesetzt werden. Dazu dienen folgende Systemvariablen (→ Kapitel *Systemmerker* (→ Seite 221)):

Systemmerker (Symbolname)	Typ	Beschreibung
LED	WORD	LED-Farbe für "LED eingeschaltet": 0x0000 = LED_GREEN (voreingestellt) 0x0001 = LED_BLUE 0x0002 = LED_RED 0x0003 = LED_WHITE 0x0004 = LED_BLACK 0x0005 = LED_MAGENTA 0x0006 = LED_CYAN 0x0007 = LED_YELLOW
LED_X	WORD	LED-Farbe für "LED ausgeschaltet": 0x0000 = LED_GREEN 0x0001 = LED_BLUE 0x0002 = LED_RED 0x0003 = LED_WHITE 0x0004 = LED_BLACK (voreingestellt) 0x0005 = LED_MAGENTA 0x0006 = LED_CYAN 0x0007 = LED_YELLOW
LED_MODE	WORD	LED-Blinkfrequenz: 0x0000 = LED_2HZ (blinkt mit 2 Hz; voreingestellt) 0x0001 = LED_1HZ (blinkt mit 1 Hz) 0x0002 = LED_05HZ (blinkt mit 0,5 Hz) 0x0003 = LED_0HZ (leuchtet dauernd mit Wert in LED) 0x0004 = LED_5HZ (blinkt mit 5 Hz)

HINWEIS

► Im Anwendungsprogramm NICHT die LED-Farbe ROT verwenden.

> Im Fehlerfall wird die LED-Farbe ROT durch das Laufzeitsystem gesetzt.
ABER: Werden die Farben und/oder Blinkmodi im Anwendungsprogramm geändert, gilt die obige Tabelle der Voreinstellung nicht mehr.

3.3 Schnittstellen-Beschreibung

Inhalt

14098

3.3.1 Serielle Schnittstelle

14099

Dieses Gerät bietet eine serielle Schnittstelle.

Grundsätzlich kann die serielle Schnittstelle mit folgenden Funktionen genutzt werden:
- Programm-Download
- Debugging
- freie Nutzung in der Anwendung

HINWEIS

Voreingestellt steht die serielle Schnittstelle dem Anwender nicht zur Verfügung, da sie für den Programm-Download und das Debugging genutzt wird.

Setzt der Anwender das Systemmerkerbit SERIAL_MODE=TRUE, dann kann die Schnittstelle frei genutzt werden. Ein Debugging des Anwendungsprogramms ist dann nur noch über eine der 4 CAN-Schnittstellen oder über USB möglich.

Anschlüsse und Daten → Datenblatt

3.3.2 USB-Schnittstelle

14100

Dieses Gerät bietet eine USB-Schnittstelle für den Programm-Download und das Debugging.

Anschlüsse und Daten → Datenblatt
USB-Treiber auf dem PC installieren → Montageanleitung / Betriebsanleitung

Einstellungen in CODESYS für [Online] > [Kommunikationsparameter...] via USB:

Gerät	Laufzeitsystem-Version	Parameter	Wert
CR0032	< V03.00.00	Baudrate	115200
CR0032	≥ V03.00.01	Baudrate	4800...57600
CR0033, CR0133	≤ V02.00.01	Baudrate	115200
CR0033, CR0133	≥ V02.00.02	Baudrate	4800...57600
CR0232, CR0233	alle	Baudrate	115200
CR0234, CR0235	alle	Baudrate	4800...57600
CR7n32	≤ V01.00.04	Baudrate	115200
CR7n32	≥ V01.00.05	Baudrate	4800...57600
CR0n3n, CR7n32	alle	Motorola byteorder	No
CR0n3n, CR7n32	alle	Flow Control	On

3.3.3 CAN-Schnittstellen

Inhalt

14101

Anschlüsse und Daten → Datenblatt

CAN: Schnittstellen und Protokolle

13820
14587

Die Geräte werden je nach Aufbau der Hardware mit mehreren CAN-Schnittstellen ausgerüstet. Grundsätzlich können alle Schnittstellen unabhängig voneinander mit folgenden Funktionen genutzt werden:

- Layer 2: CAN auf Ebene 2 (→ Kapitel *Bausteine: CAN Layer 2* (→ Seite 73))
- CANopen-Master (→ Kapitel *Bausteine: CANopen-Master* (→ Seite 83))
- CANopen-Slave (→ Kapitel *Bausteine: CANopen-Slave* (→ Seite 93))
- CANopen-Netzwerkvariablen (via CODESYS)
- SAE J1939 (für Antriebsmanagement, → Kapitel *Bausteine: SAE J1939* (→ Seite 106))
- Buslast-Erkennung
- Errorframe-Zähler
- Download-Schnittstelle
- 100 % Buslast ohne Paketverlust

11793

In diesem **ecomat*mobile***-Gerät sind folgende CAN-Schnittstellen und CAN-Protokolle verfügbar:

CAN-Schnittstelle	CAN 1	CAN 2	CAN 3	CAN 4
voreingestellte Download-ID	ID 127	ID 126	ID 125	ID 124
CAN-Protokolle	CAN Layer 2	CAN Layer 2	CAN Layer 2	CAN Layer 2
	CANopen	CANopen	CANopen	CANopen
	SAE J1939	SAE J1939	SAE J1939	SAE J1939

Standard-Baudrate = 125 kBit/s

ℹ Welche CANopen-fähige Schnittstelle mit welchem CANopen-Protokoll arbeitet, entscheidet die Reihenfolge, mit der Sie in der Steuerungskonfiguration die Unterelemente anhängen:
CODESYS > [Steuerungskonfiguration] > [CR0033 Configuration Vxx] > [Unterelement anhängen] > [CANopen Master] oder [CANopen Slave]

3.4 Software

Inhalt

14107

3.4.1 Software-Module für das Gerät

Inhalt

14110

Die Software in diesem Gerät setzt wie folgt auf der Hardware auf:

Software-Modul	Anwender kann das Modul ändern?	womit?
Anwendungsprogramm mit Bibliotheken	ja	CODESYS, MaintenanceTool
Laufzeitsystem (LZS) *)	Upgrade ja Downgrade ja	MaintenanceTool
Bootloader	nein	---
(Hardware)	nein	---

*) Die Laufzeitsystem-Versionsnummer muss der Target-Versionsnummer in der CODESYS-Zielsystemeinstellung entsprechen!
→ Kapitel *Target einrichten* (→ Seite 54)

Nachfolgend beschreiben wir diese Software-Module:

Bootloader

14111

Im Auslieferungszustand enthalten **ecomat*mobile***-Controller nur den Bootloader.
Der Bootloader ist ein Startprogramm, mit dem das Laufzeitsystem und das Anwendungsprogramm auf dem Gerät nachgeladen werden können.
Der Bootloader enthält Grundroutinen...

- zur Kommunikation der Hardware-Module untereinander,
- zum Nachladen des Laufzeitsystems.

Der Bootloader ist das erste Software-Modul, das im Gerät gespeichert sein muss.

Laufzeitsystem

14112

Grundprogramm im Gerät, stellt die Verbindung her zwischen der Hardware des Gerätes und dem Anwendungsprogramm.
→ Kapitel *Software-Module für das Gerät* (→ Seite 37)

Im Auslieferungszustand ist im Normalfall kein Laufzeitsystem im Controller geladen (LED blinkt grün mit 5 Hz). In diesem Betriebszustand ist nur der Bootloader aktiv. Dieser stellt die minimalen Funktionen für den Laufzeitsystem-Ladevorgang zur Verfügung, u.a. die Unterstützung der Schnittstellen (z.B. CAN).

Der Laufzeitsystem-Download muss im Normalfall nur einmalig durchgeführt werden. Das Anwendungsprogramm kann anschließend (auch mehrmals) in den Controller geladen werden, ohne das Laufzeitsystem zu beeinflussen.

Das Laufzeitsystem wird zusammen mit dieser Dokumentation auf einem separaten Datenträger zur Verfügung gestellt. Zusätzlich kann auch die aktuelle Version von der Homepage der **ifm electronic gmbh** heruntergeladen werden:
→ www.ifm.com > Land wählen > [Service] > [Download]

Anwendungsprogramm

14118

Software, die speziell für die Anwendung vom Hersteller in die Maschine programmiert wird. Die Software enthält üblicherweise logische Sequenzen, Grenzwerte und Ausdrücke zum Steuern der entsprechenden Ein- und Ausgänge, Berechnungen und Entscheidungen.

8340

⚠ WARNUNG

Für die sichere Funktion der Anwendungsprogramme, die vom Anwender erstellt werden, ist dieser selbst verantwortlich. Bei Bedarf muss er zusätzlich entsprechend der nationalen Vorschriften eine Abnahme durch entsprechende Prüf- und Überwachungsorganisationen durchführen lassen.

Bibliotheken

14117

ifm electronic bietet passend für jedes Gerät eine Reihe von Bibliotheken (`*.LIB`) an, die Programmmodule für das Anwendungsprogramm enthalten. Beispiele:

Bibliothek	Verwendung
`ifm_CR0033_Vxxyyzz.LIB`	gerätespezifische Bibliothek Muss immer im Anwendungsprogramm enthalten sein!
`ifm_CR0033_CANopenxMaster_Vxxyyzz.LIB` x = 1...4 = Nummer der CAN-Schnittstelle	(optional) wenn eine CAN-Schnittstelle des Geräts als CANopen-Master betrieben werden soll
`ifm_CR0033_CANopenxSlave_Vxxyyzz.LIB` x = 1...4 = Nummer der CAN-Schnittstelle	(optional) wenn eine CAN-Schnittstelle des Geräts als CANopen-Slave betrieben werden soll
`ifm_CR0033_J1939_Vxxyyzz.LIB`	(optional) wenn eine CAN-Schnittstelle des Geräts mit einer Motorsteuerung kommunizieren soll

→ Kapitel *ifm-Bibliotheken für das Gerät CR0033* (→ Seite 68)

3.4.2 Programmierhinweise für CODESYS-Projekte

Inhalt

7426

Hier erhalten Sie Tipps zum Programmieren des Geräts.

► Beachten Sie die Hinweise im CODESYS-Programmierhandbuch
→ www.ifm.com > Land wählen > [Datenblattsuche] > CR0033 > [Betriebsanleitungen],
→ **ecomat*mobile***-DVD "Software, tools and documentation".

FB, FUN, PRG in CODESYS

8473

In CODESYS unterscheiden wir folgende Typen von Bausteinen (POUs):

FB = function block = Funktionsbaustein
- Ein FB kann mehrere Eingänge und mehrere Ausgänge haben.
- Ein FB darf in einem Projekt mehrmals aufgerufen werden.
- Für jeden Aufruf muss eine Instanz deklariert werden.
- Erlaubt: Im FB aufrufen von FB und FUN.

FUN = function = Funktion
- Eine Funktion kann mehrere Eingänge, aber nur einen Ausgang haben.
- Der Ausgang ist vom gleichen Datentyp wie die Funktion selbst.

PRG = program = Programm
- Ein PRG kann mehrere Eingänge und mehrere Ausgänge haben.
- Ein PRG darf in einem Projekt nur einmal aufgerufen werden.
- Erlaubt: im PRG aufrufen von PRG, FB und FUN.

HINWEIS

Funktionsbausteine dürfen NICHT in Funktionen aufgerufen werden!
Sonst: Bei der Ausführung stürzt das Anwendungsprogramm ab.

Alle Bausteine (POUs) dürfen NICHT rekursiv aufgerufen werden, auch nicht indirekt!

Eine IEC-Anwendung darf maximal 8000 Bausteine (POU) enthalten!

Hintergrund:

Alle Variablen von Funktionen...
- werden beim Aufruf initialisiert und
- werden nach der Rückkehr zum Aufrufer ungültig.

Funktionsbausteine haben 2 Aufrufe:
- einen Initialisierungsaufruf und
- den eigentlichen Aufruf, um irgend etwas zu tun.

Folglich heißt das für den FB-Aufruf in einer Funktion:
- jedesmal erfolgt ein zusätzlicher Initialisierungsaufruf und
- die Daten des letzten Aufrufs gehen verloren.

Zykluszeit beachten!

8006

Bei den frei programmierbaren Geräten aus der Controller-Familie ecomat*mobile* stehen in einem großen Umfang Bausteine zur Verfügung, die den Einsatz der Geräte in den unterschiedlichsten Anwendungen ermöglichen.

Da diese Bausteine je nach Komplexität mehr oder weniger Systemressourcen belegen, können nicht immer alle Bausteine gleichzeitig und mehrfach eingesetzt werden.

ACHTUNG

Gefahr von zu trägem Verhalten des Geräts!
Zykluszeit darf nicht zu lang werden!

► Beim Erstellen des Anwendungsprogramms die oben aufgeführten Empfehlungen beachten und durch Austesten überprüfen.

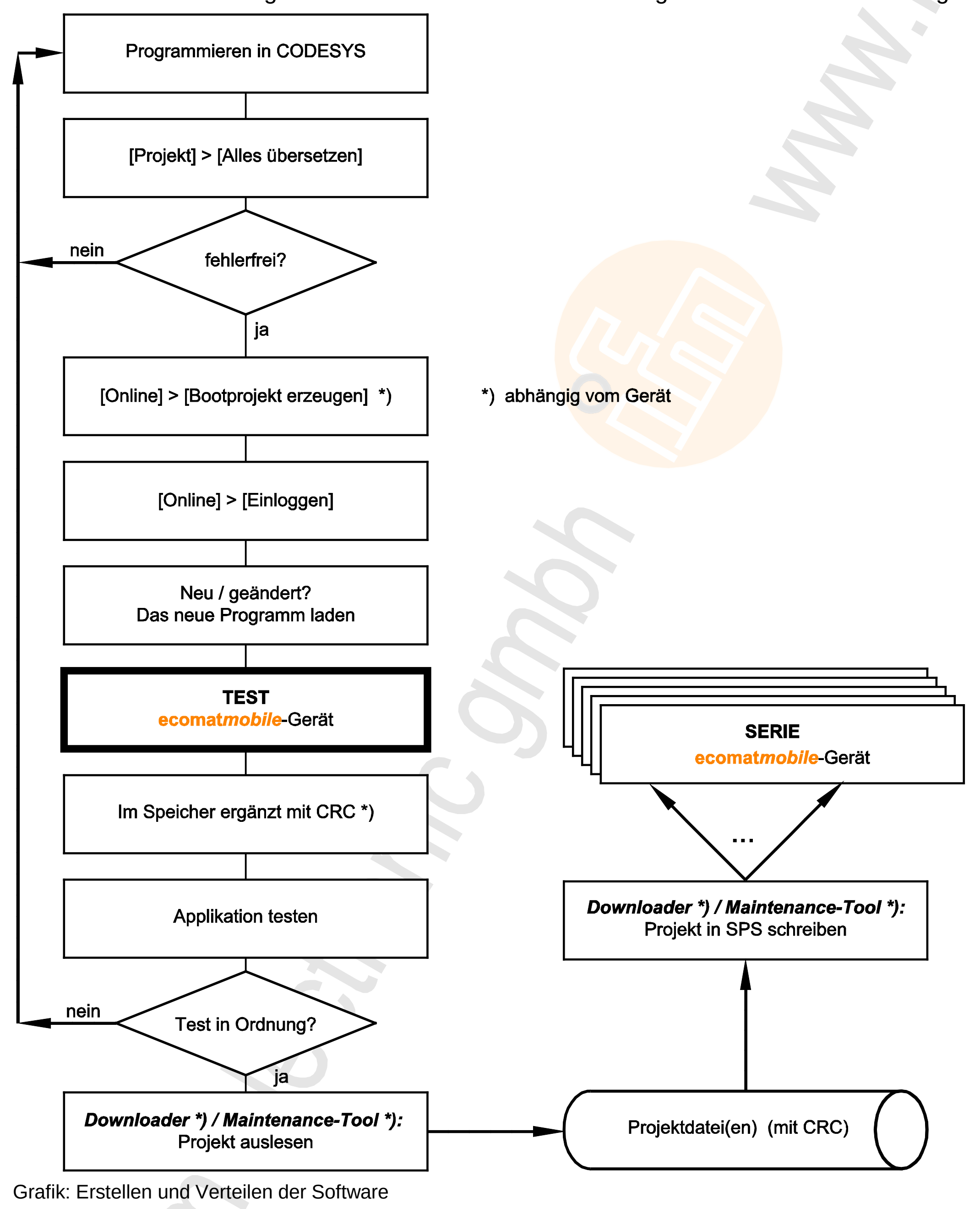
► Bei Bedarf durch Neustrukturieren der Software und des Systemaufbaus die Zykluszeit vermindern.

Anwendungsprogramm erstellen

8007

Das Anwendungsprogramm wird mit dem Programmiersystem CODESYS erstellt und während der Programmentwicklung mehrfach zum Testen in die Steuerung geladen:
In CODESYS: [Online] > [Einloggen] > das neue Programm laden.

Für jeden derartigen Download via CODESYS wird dazu der Quellcode neu übersetzt. Daraus resultiert, dass auch jedes Mal im Speicher der Steuerung eine neue Prüfsumme gebildet wird. Auch für Sicherheitssteuerungen ist dieses Verfahren bis zur Freigabe der Software zulässig.



Grafik: Erstellen und Verteilen der Software

Boot-Projekt speichern

7430

Speichern Sie im Gerät zusammen mit Ihrem Anwendungsprogramm immer auch das zugehörige Boot-Projekt! Nur so ist das Anwendungsprogramm auch nach einem Spannungsausfall im Gerät verfügbar.

> **HINWEIS**
>
> Beachten: das Boot-Projekt ist etwas größer als das eigentliche Programm.
>
> Jedoch: das Speichern des Boot-Projekts im Gerät wird scheitern, wenn das Boot-Projekt größer wird als der vorhandene IEC-Code-Speicherbereich. Nach Power-On-Reset ist das Boot-Projekt wieder gelöscht oder ungültig.

► CODESYS-Menü [Online] > [Bootprojekt erzeugen]
Dies muss auch nach jeder Änderung erneut erfolgen!

> Nach einem Neustart startet das Gerät mit dem zuletzt gespeicherten Boot-Projekt.

> Falls noch KEIN Boot-Projekt gespeichert wurde:
• das Gerät bleibt nach dem Neustart im STOP-Betrieb
• das Anwendungsprogramm ist nicht (mehr) vorhanden
• die LED leuchtet grün.

ifm-Downloader nutzen

8008

Der **ifm**-Downloader dient dem einfachen Übertragen des Programmcodes vom Programmierplatz in die Steuerung. Grundsätzlich kann jedes Anwendungsprogramm mit dem **ifm**-Downloader auf die Steuerungen kopiert werden. Vorteil: Dazu ist kein Programmiersystem mit einer CODESYS-Lizenz erforderlich.

Hier finden Sie den aktuellen **ifm**-Downloader (min. V06.18.26):
→ www.ifm.com > Land wählen > [Service] > [Download] > [Systeme für mobile Arbeitsmaschinen]
ecomat*mobile*-DVD "Software, tools and documentation" im Register "R360 tools [D/E]"

ifm-Maintenance-Tool nutzen

8492

Das **ifm**-Maintenance-Tool dient dem einfachen Übertragen des Programmcodes vom Programmierplatz in das Gerät. Grundsätzlich kann jedes Anwendungsprogramm mit dem **ifm**-Maintenance-Tool auf die Geräte kopiert werden. Vorteil: Dazu ist kein Programmiersystem mit einer CODESYS-Lizenz erforderlich.

Hier finden Sie das aktuelle **ifm**-Maintenance-Tool:
→ www.ifm.com > Land wählen > [Service] > [Download] > [Systeme für mobile Arbeitsmaschinen]
→ **ecomat*mobile***-DVD "Software, tools and documentation" im Register "R360 tools [D/E]"

3.4.3 Betriebszustände

Inhalt

14120

Nach Anlegen der Versorgungsspannung kann sich das **ecomat*mobile***-Gerät in einem von fünf möglichen Betriebszuständen befinden:

- BOOTLOADER
- INIT
- STOP
- RUN
- SYSTEM STOP (nach ERROR STOP)

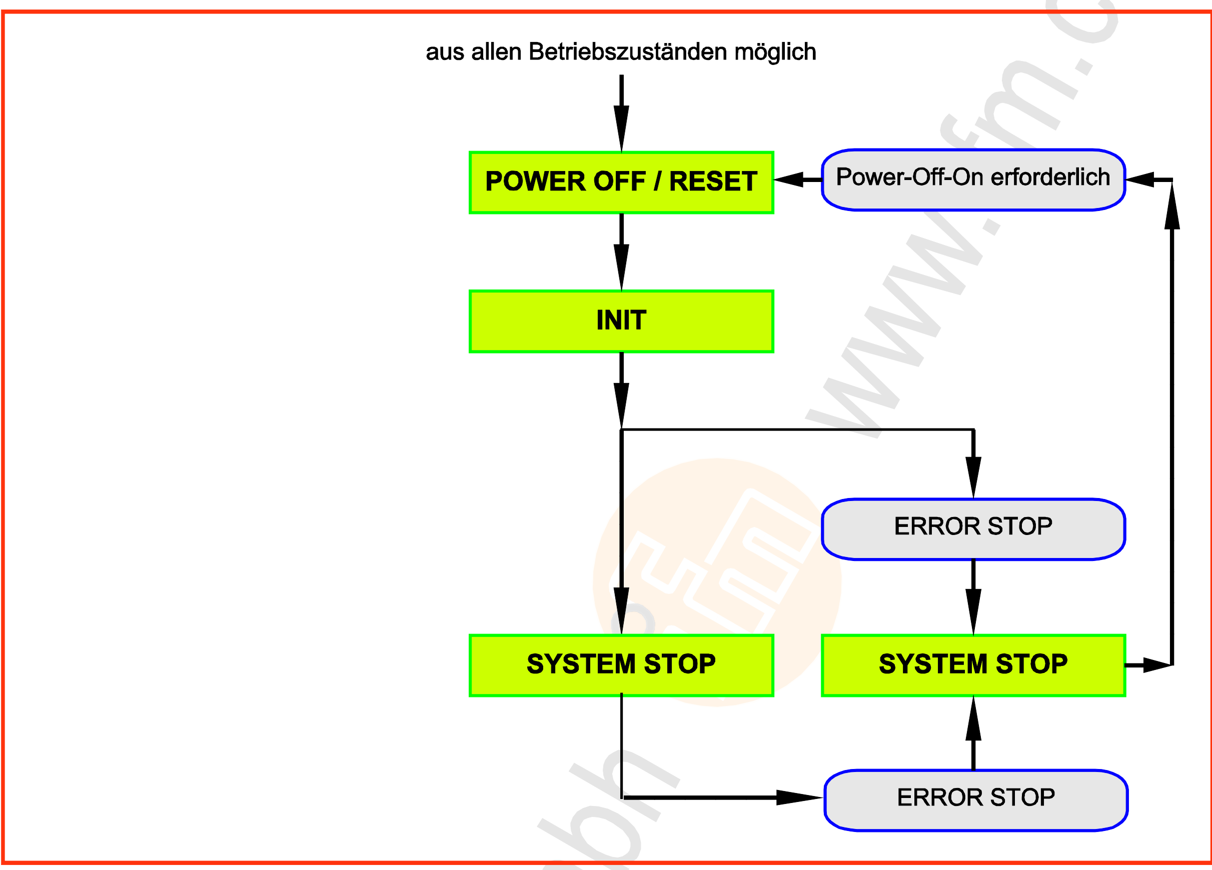
Betriebszustände

19217

Grafik: Betriebszustände (hier: Laufzeitsystem ist nicht verfügbar)

Betriebszustände: Anwendungsprogramm nicht verfügbar

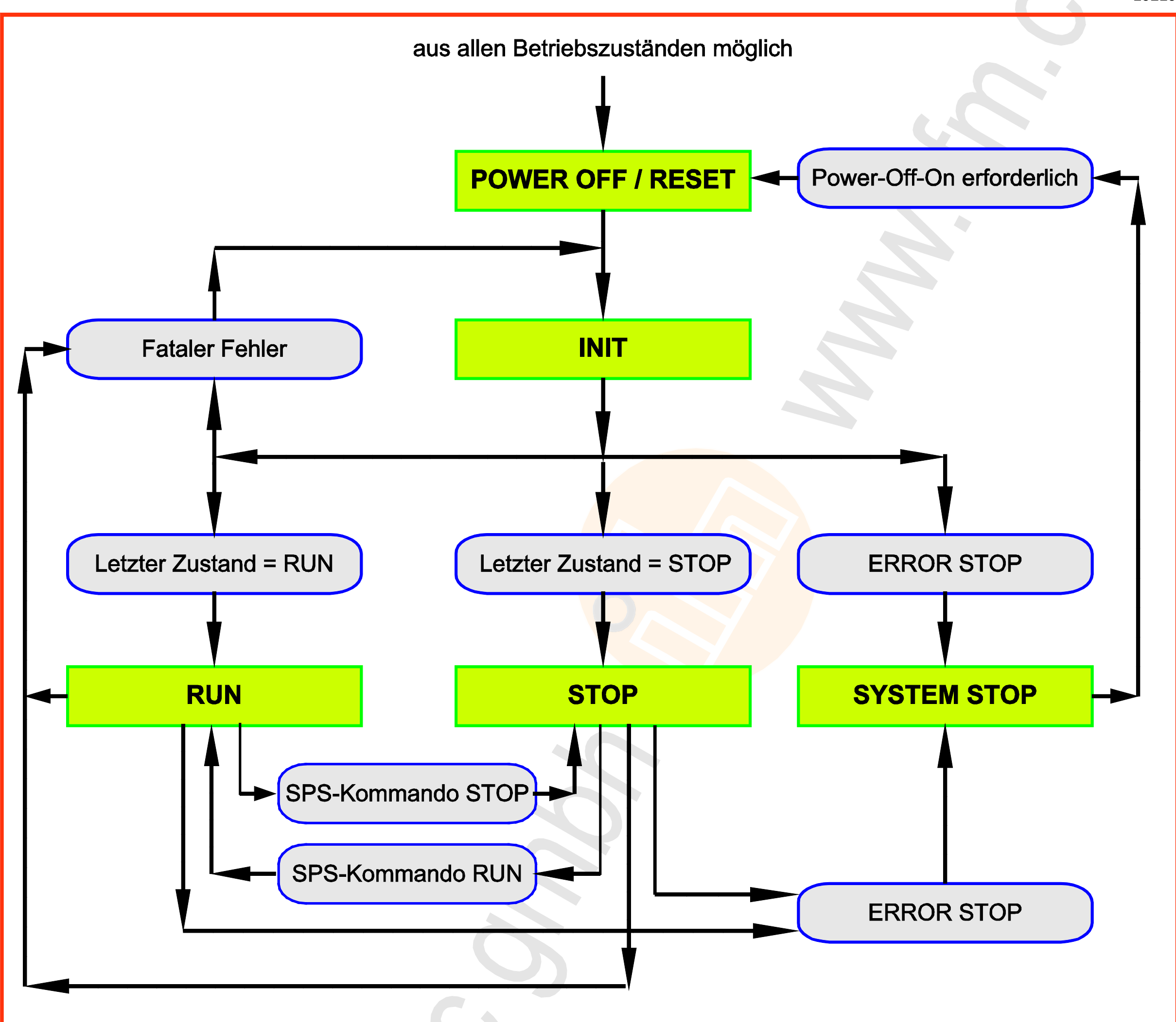
19218



Grafik: Betriebszustände (hier: Anwendungsprogramm ist nicht verfügbar)

Betriebszustände: Anwendungsprogramm verfügbar

19219



Grafik: Betriebszustände (hier: Anwendungsprogramm ist verfügbar)

Bootloader-Zustand

1080

Es wurde kein Laufzeitsystem geladen. Der **ecomat*mobile***-Controller befindet sich im Bootloader-Zustand. Vor dem Laden des Anwendungsprogramms muss ein Laufzeitsystem-Download durchgeführt werden.

> Die LED blinkt grün (5 Hz).

INIT-Zustand (Reset)

1076

Voraussetzung: ein gültiges Laufzeitsystem ist installiert.

Dieser Zustand wird nach jedem Power-On-Reset durchlaufen:

> Das Laufzeitsystem wird initialisiert.

> Verschiedene Checks werden durchgeführt, z.B. Warten auf gültige Versorgungsspannung.

> Dieser nur temporäre Zustand wird vom RUN- oder STOP-Zustand abgelöst.

> Die LED leuchtet gelb.

Wechsel aus diesem Zustand in einen der folgenden Zustände möglich:
• RUN
• STOP

STOP-Zustand

1078

Dieser Zustand wird in folgenden Fällen erreicht:

- Aus dem RESET-Zustand, wenn:
  • kein Anwendungsprogramm ist geladen oder
  • der letzte Zustand vor dem RESET-Zustand war der STOP-Zustand
- Aus dem RUN-Zustand durch das STOP-Kommando
  • nur bei Betriebsmodus = TEST (→ Kapitel *TEST-Betrieb* (→ Seite 48))

> Die LED leuchtet grün.

RUN-Zustand

1077

Dieser Zustand wird in folgenden Fällen erreicht:

- Aus dem RESET-Zustand, wenn:
  • der letzte Zustand vor dem RESET-Zustand war der RUN-Zustand
- Aus dem STOP-Zustand durch das RUN-Kommando
  • nur bei Betriebsmodus = TEST (→ Kapitel *TEST-Betrieb* (→ Seite 48))

> Die LED blinkt grün (2 Hz).

SYSTEM-STOP-Zustand

19222

In diesen Zustand fällt der **ecomat*mobile***-Controller, wenn ein nicht tolerierbarer Fehler (ERROR STOP) festgestellt wurde. Dieser Zustand kann nur durch einen Power-Off-On-Reset verlassen werden.

> Die LED leuchtet rot.

3.4.4 Betriebsmodi

1083

Unabhängig von den Betriebszuständen kann der Controller in verschiedenen Betriebsmodi betrieben werden.

TEST-Betrieb

1084

ACHTUNG

Verlust der gespeicherten Software möglich!
Im Test-Betrieb besteht kein Schutz der gespeicherten Laufzeitsystem- und Anwendungs-Software.

HINWEIS

► Erst NACH dem Anschließen des OPC-Client den TEST-Anschluss mit der Versorgungsspannung verbinden!

> Ansonsten tritt ein fataler Fehler auf.

Dieser Betriebsmodus wird durch Anlegen von Versorgungsspannung am Test-Eingang erreicht (→ Montageanleitung > Kapitel "Technische Daten" > Kapitel "Anschlussbelegung").

Jetzt kann der Controller im RUN- oder STOP-Zustand Kommandos über eine der Schnittstellen entgegennehmen und z.B. mit dem Programmiersystem kommunizieren.

Nur im TEST-Betrieb ist ein Software-Download im Controller möglich.

Über den Merker TEST kann der Zustand vom Anwendungsprogramm abgefragt werden.

Zusammenfassung Test-Eingang ist aktiv:
- Programmiermodus ist freigeben
- Software-Download ist möglich
- Zustand des Anwendungsprogramms ist abfragbar
- kein Schutz der gespeicherten Software möglich

SERIAL_MODE

2548

Die serielle Schnittstelle steht für den Datenaustausch in der Anwendung zur Verfügung. Ein Debugging des Anwendungsprogramms ist dann nur noch über alle 4 CAN-Schnittstellen oder über USB möglich.

Diese Funktion ist standardmäßig abgeschaltet (FALSE). Über den Merker SERIAL_MODE kann der Zustand über das Anwendungsprogramm oder das Programmiersystem gesteuert und abgefragt werden.

(→ Kapitel *Bausteine: serielle Schnittstelle* (→ Seite 118))

DEBUG-Modus

1086

Wird der Eingang DEBUG von *SET_DEBUG* (→ Seite 215) auf TRUE gesetzt, kann z.B. das Programmiersystem oder der Downloader mit dem Gerät kommunizieren und Systemkommandos ausführen (z.B. für Servicefunktionen über das GSM-Modem CANremote).

Ein Software-Download ist in dieser Betriebsart nicht möglich, da der Test-Eingang (→ Kapitel *TEST-Betrieb* (→ Seite 48)) nicht mit Versorgungsspannung verbunden wird.

3.4.5 Leistungsgrenzen des Geräts

7358



Leistungsgrenzen des Geräts beachten! → Datenblatt

Verhalten des Watchdog

11786

Ein Watchdog überwacht in diesem Gerät die Programmlaufzeit der CODESYS-Anwendung.
Wird die maximale Watchdog-Zeit (ca. 100 ms) überschritten:
> das Gerät führt einen Reset durch und startet neu
Zu erkennen im Merker LAST_RESET.

CODESYS-Funktionen

2254

Folgende Grenzen sollten Sie berücksichtigen:

- Bis zu 2048 Bausteine (PB, FB...) werden unterstützt.
- Für Anwender verfügbare Merker → Kapitel *Verfügbarer Speicher* (→ Seite 14).
  Beschreibung der Retain-Merker → bei den jeweiligen FBs.

4 Konfigurationen

Inhalt

18065
1016

Die in den jeweiligen Montage- und Installationsanweisungen oder dem *Anhang* (→ Seite 221) dieser Dokumentation beschriebenen Gerätekonfigurationen stehen als Standardgeräte (Lagerware) zur Verfügung. Diese decken bei den meisten Anwendungen die geforderten Spezifikationen ab.

Entsprechend den Kundenanforderungen bei Serieneinsatz ist es aber auch möglich, dass andere Gerätekonfigurationen z.B. hinsichtlich der Zusammenstellung der Ein- und Ausgänge und der Ausführung der Analogkanäle eingesetzt werden.

16420

HINWEIS

Diese Anleitung gilt für das Gerät ohne und mit integriertem E/A-Modul.

► In beiden Fällen die Steuerungskonfiguration unbedingt für das Gerät CR0033 einrichten!

Die Beschreibung zum integrierten E/A-Modul finden Sie hier:
→ Kapitel *Integriertes E/A-Modul: Beschreibung* (→ Seite 240) im Anhang dieser Dokumentation.

4.1 Laufzeitsystem einrichten

Inhalt

14091

4.1.1 Laufzeitsystem neu installieren

14092
2733

Im Auslieferungszustand ist im Normalfall kein Laufzeitsystem im Gerät geladen (LED blinkt grün mit 5 Hz). In diesem Betriebszustand ist nur der Bootloader aktiv. Dieser stellt die minimalen Funktionen für den Laufzeitsystem-Ladevorgang zur Verfügung, u.a. die Unterstützung der Schnittstellen (z.B. RS232, CAN).

Der Laufzeitsystem-Download muss im Normalfall nur einmalig durchgeführt werden. Das Anwendungsprogramm kann anschließend (auch mehrmals) in das Gerät geladen werden, ohne das Laufzeitsystem zu beeinflussen.

Das Laufzeitsystem wird zusammen mit dieser Dokumentation auf einem separaten Datenträger zur Verfügung gestellt. Zusätzlich kann auch die aktuelle Version von der Homepage der **ifm electronic gmbh** heruntergeladen werden:
→ www.ifm.com > Land wählen > [Service] > [Download]

2689

HINWEIS

Es müssen immer die zum gewählten Target passenden Software-Stände zum Einsatz kommen:
- des Laufzeitsystems (`ifm_CR0033_Vxxyyzz.H86`),
- der Steuerungskonfiguration (`ifm_CR0033_Vxx.CFG`),
- der Gerätebibliothek (`ifm_CR0033_Vxxyyzz.LIB`) und
- der weiteren Dateien

V	Version
xx: 00...99	Versionsnummer
yy: 00...99	Release-Nummer
zz: 00...99	Patch-Nummer

Dabei müssen der Basisdateiname (z.B. "`CR0033`") und die Software-Versionsnummer "xx" (z.B. "`02`") überall den gleichen Wert haben! Andernfalls geht das Gerät in den STOP-Zustand.

Die Werte für "yy" (Release-Nummer) und "zz" (Patch-Nummer) müssen **nicht** übereinstimmen.

4368

Folgende Dateien müssen ebenfalls geladen sein:
- die zum Projekt erforderlichen internen Bibliotheken (in IEC 61131 erstellt),
- die Konfigurationsdateien (`*.CFG`)
- und die Target-Dateien (`*.TRG`).

Es kann vorkommen, dass das Zielsystem mit Ihrer aktuell installierten Version von CODESYS nicht oder nur teilweise programmiert werden kann. Im diesem Fall wenden Sie sich bitte an den technischen Support der **ifm electronic gmbh**.

Das Laufzeitsystem wird mit dem eigenständigen Programm "**ifm**-Downloader" in das Gerät übertragen. (Der **ifm**-Downloader und dessen Dokumentation befindet sich auf der **ecomat*mobile***-DVD "Software, tools and documentation" oder kann bei Bedarf von der **ifm**-Homepage heruntergeladen werden: → www.ifm.com > Land wählen > [Service] > [Download]).

Das Anwendungsprogramm wird im Normalfall über das Programmiersystem in das Gerät geladen. Es kann aber ebenfalls mit dem **ifm**-Downloader geladen werden, wenn es zuvor aus dem Gerät ausgelesen wurde (→ Upload).

4.1.2 Laufzeitsystem aktualisieren

13269

Auf dem Gerät ist bereits ein älteres Laufzeitsystem installiert. Nun möchten Sie das Laufzeitsystem auf dem Gerät aktualisieren?

14158

ACHTUNG

Gefahr von Datenverlust!

Beim Löschen oder Aktualisieren des Laufzeitsystems werden alle Daten und Programme auf dem Gerät gelöscht.

► Alle erforderlichen Daten und Programme sichern, bevor das Laufzeitsystem gelöscht oder aktualisiert wird!

3084

Immer, wenn es zu wesentlichen Verbesserungen in der Betriebsystem-Software oder des CODESYS-Laufzeitsystems kommt, gibt **ifm** davon eine neue Version heraus. Die Versionen werden fortlaufend durchnummeriert (V01, V02, V03, ...).

Welche neuen Zusatzfunktionen die neue Softwareversion enthält, entnehmen Sie bitte der jeweiligen Dokumentation. Beachten Sie, ob in der Dokumentation auf besondere Anforderungen an die Hardware-Version hingewiesen wird.

Wenn Sie im Besitz eines Gerätes mit einer älteren Version sind und wenn die Bedingungen für die Hardware und Ihr Projekt stimmen, können Sie Ihr Gerät durch Aktualisieren der Software auf den neuen Software-Stand bringen.

Prinzipiell gelten für diesen Vorgang die gleichen Hinweise, wie zuvor im Kapitel 'Laufzeitsystem neu installieren' gegeben wurden.

4.1.3 Installation verifizieren

14407
14406

► Nach dem Laden des Laufzeitsystems in die Steuerung:
  • Prüfen, ob das Laufzeitsystem korrekt übertragen wurde!
  • Prüfen, ob sich das richtige Laufzeitsystem auf der Steuerung befindet!

► 1. Prüfung:
mit dem **ifm**-Downloader oder mit dem Maintenance-Tool prüfen, ob die richtige Laufzeitsystem-Version geladen wurde:
  • Name, Version und die CRC des Laufzeitsystems im Gerät auslesen!
  • Diese Daten manuell mit den Soll-Daten vergleichen!

► 2. Prüfung (optional):
Im Anwendungsprogramm prüfen, ob die richtige Laufzeitsystem-Version geladen wurde:
  • Name und die Version des Laufzeitsystems im Gerät auslesen!
  • Diese Daten mit fest vorgegebenen Werten vergleichen!
Zum Auslesen der Daten dient folgender FB:

GET_IDENTITY (→ Seite 214)	liest die im Gerät gespeicherten spezifischen Kennungen: • Hardware-Name und Hardware-Version des Geräts • Name des Laufzeitsystems im Gerät • Version und Ausgabe des Laufzeitsystems im Gerät • Name der Anwendung (wurde zuvor mit *SET_IDENTITY* (→ Seite 216) gespeichert) • Seriennummer des Geräts

4.2 Programmiersystem einrichten

Inhalt

3968

4.2.1 Programmiersystem manuell einrichten

Inhalt

3963

Target einrichten

2687
11379

Beim Erstellen eines neuen Projektes in CODESYS muss die dem Gerät entsprechende Target-Datei geladen werden.

► Im Dialog-Fenster [Zielsystem Einstellungen] im Menü [Konfiguration] die gewünschte Target-Datei wählen.

> Die Target-Datei stellt für das Programmiersystem die Schnittstelle zur Hardware her.

> Gleichzeitig mit Wahl des Targets werden automatisch einige wichtige Bibliotheken und die Steuerungskonfiguration geladen.

► Bei Bedarf im Fenster [Zielsystem Einstellungen] > Reiter [Netzfunktionen] > [Parameter-Manager unterstützen] und / oder [Netzvariablen unterstützen] aktivieren.

► Bei Bedarf geladene (3S-)Bibliotheken wieder entfernen oder durch weitere (ifm-)Bibliotheken ergänzen.

► Immer die passende Geräte-Bibliothek `ifm_CR0033_Vxxyyzz.LIB` manuell ergänzen!

2689

HINWEIS

Es müssen immer die zum gewählten Target passenden Software-Stände zum Einsatz kommen:
- des Laufzeitsystems (`ifm_CR0033_Vxxyyzz.H86`),
- der Steuerungskonfiguration (`ifm_CR0033_Vxx.CFG`),
- der Gerätebibliothek (`ifm_CR0033_Vxxyyzz.LIB`) und
- der weiteren Dateien

V	Version
xx: 00...99	Versionsnummer
yy: 00...99	Release-Nummer
zz: 00...99	Patch-Nummer

Dabei müssen der Basisdateiname (z.B. "`CR0033`") und die Software-Versionsnummer "xx" (z.B. "`02`") überall den gleichen Wert haben! Andernfalls geht das Gerät in den STOP-Zustand.

Die Werte für "yy" (Release-Nummer) und "zz" (Patch-Nummer) müssen **nicht** übereinstimmen.

4368

Folgende Dateien müssen ebenfalls geladen sein:
- die zum Projekt erforderlichen internen Bibliotheken (in IEC 61131 erstellt),
- die Konfigurationsdateien (`*.CFG`)
- und die Target-Dateien (`*.TRG`).

Es kann vorkommen, dass das Zielsystem mit Ihrer aktuell installierten Version von CODESYS nicht oder nur teilweise programmiert werden kann. Im diesem Fall wenden Sie sich bitte an den technischen Support der **ifm electronic gmbh**.

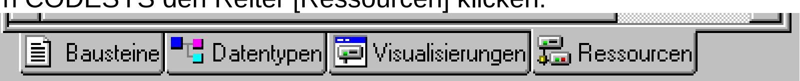
Steuerungskonfiguration aktivieren (z.B. CR0033)

15824

Bei der Konfiguration des Programmiersystems (→ vorheriger Abschnitt) erfolgte automatisch auch die Steuerungskonfiguration.

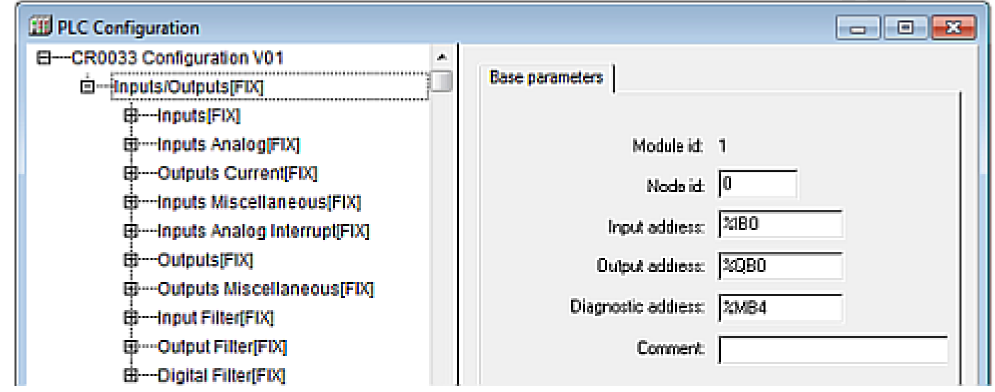
► Den Punkt [Steuerungskonfiguration] erreicht man über den Reiter [Ressourcen].
Mit Doppelklick auf den Punkt [Steuerungskonfiguration] öffnet sich das entsprechende Fenster.

► In CODESYS den Reiter [Ressourcen] klicken:



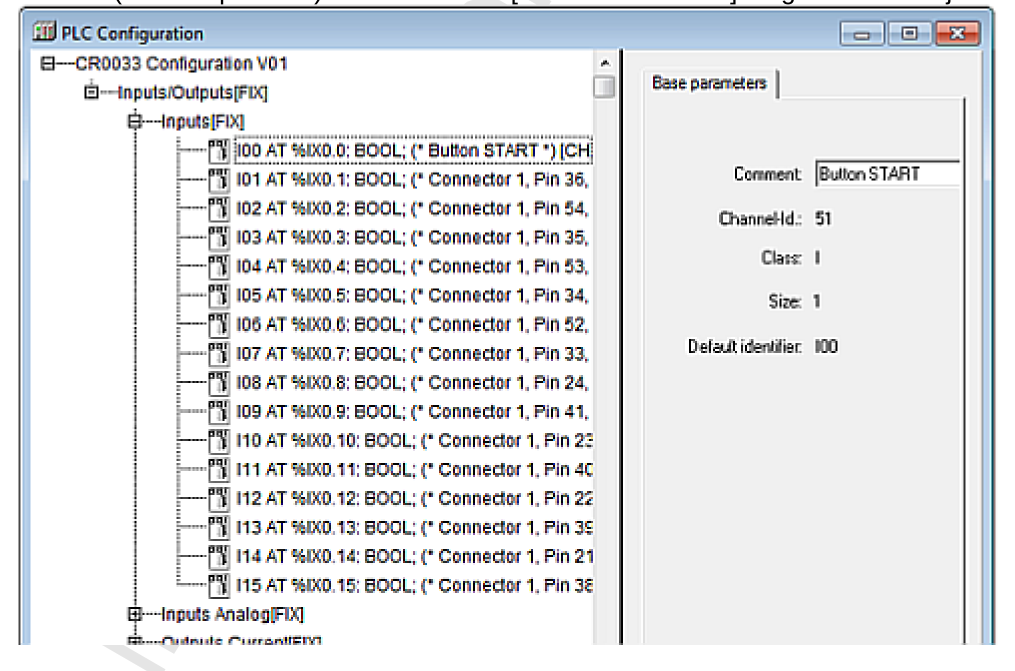
► In der linken Spalte Doppelklick auf [Steuerungskonfiguration]

> Anzeige der aktuellen Steuerungskonfiguration (Beispiel → folgendes Bild):



Durch die Konfiguration ist für den Anwender in der Programmumgebung Folgendes verfügbar:

- alle wichtigen System- und Fehlermerker
  Je nach Anwendung und Anwendungsprogramm müssen diese Merker bearbeitet und ausgewertet werden. Der Zugriff erfolgt über deren symbolischen Namen.
- die Struktur der Ein- und Ausgänge
  Diese können im Fenster [Steuerungskonfiguration] (→ Bild unten) direkt symbolisch bezeichnet werden (sehr empfohlen!) und stehen als [Globale Variablen] im gesamten Projekt zur Verfügung.



4.2.2 Programmiersystem über Templates einrichten

13745

ifm bietet vorgefertigte Templates (Programm-Vorlagen), womit Sie das Programmiersystem schnell, einfach und vollständig einrichten können.

970

Beim Installieren der **ecomat*mobile***-DVD "Software, tools and documentation" wurden auch Projekte mit Vorlagen auf Ihrem Computer im Programmverzeichnis abgelegt:
`…\ifm electronic\CoDeSys V…\Projects\Template_DVD_V…`

► Die gewünschte dort gespeicherte Vorlage in CODESYS öffnen mit:
[Datei] > [Neu aus Vorlage…]

> CODESYS legt ein neues Projekt an, dem der prinzipielle Programmaufbau entnommen werden kann. Es wird dringend empfohlen, dem gezeigten Schema zu folgen.

4.3 Funktionskonfiguration, allgemein

Inhalt

3971

4.3.1 Konfiguration der Ein- und Ausgänge (Voreinstellung)

2249

- Alle Ein-/Ausgänge sind im Auslieferungszustand im Binär-Modus (plus-schaltend!).
- Die Diagnosefunktion ist nicht aktiv.
- Der Überlastschutz ist aktiv.

4.3.2 Systemvariablen

13519
15576

Alle Systemvariablen (→ Kapitel *Systemmerker* (→ Seite 221)) liegen auf festen, nicht verschiebbaren Adressen.

> Zur Anzeige und Verarbeitung eines Watchdog-Fehlers oder Ursachen eines Neustarts wird die Systemvariable LAST_RESET gesetzt.

4.4 Funktionskonfiguration der Ein- und Ausgänge

Inhalt

1394

Bei bestimmten Ein- und Ausgängen sind zusätzliche Diagnosefunktionen aktivierbar. Damit kann das jeweilige Ein- und Ausgangssignal überwacht werden und im Fehlerfall kann das Anwendungsprogramm darauf reagieren.

Je nach Ein- und Ausgang müssen bei der Nutzung der Diagnose bestimmte Randbedingungen beachtet werden:

► Anhand des Datenblattes prüfen, für welche Ein- und Ausgänge des Geräts welche Diagnosemöglichkeit zur Verfügung steht!

- Zur Konfiguration der Ein- und Ausgänge sind in den Gerätebibliotheken (`ifm_CR0033_Vxxyyzz.LIB`) Konstanten vordefiniert (z.B. IN_DIGITAL_H). Ausführliche Angaben → Kapitel *Mögliche Betriebsarten Ein-/Ausgänge* (→ Seite 232).

4.4.1 Eingänge konfigurieren

Inhalt

3973

Zulässige Betriebsarten → Kapitel *Mögliche Betriebsarten Ein-/Ausgänge* (→ Seite 232)

Sicherheitshinweise zu Reed-Relais

7348

Beim Einsatz von nichtelektronischen Schaltern Folgendes beachten:

> ⓘ Kontakte von Reed-Relais können (reversibel) verkleben, wenn sie ohne Vorwiderstand an den Geräte-Eingängen angeschlossen werden.

- ► **Abhilfe:** Vorwiderstand zum Reed-Relais installieren:
  Vorwiderstand = max. Eingangsspannung / zulässiger Strom im Reed-Relais
  Beispiel: 32 V / 500 mA = 64 Ohm
- ► Der Vorwiderstand darf 5 % des Eingangswiderstands RE des Geräte-Eingangs (→ Datenblatt) nicht überschreiten. Sonst wird das Signal nicht als TRUE erkannt.
  Beispiel:
  RE = 3 000 Ohm
  ⇨ max. Vorwiderstand = 150 Ohm

Software-Filter der Eingänge konfigurieren

6883

Über die Systemvariablen Ixx_FILTER kann ein Software-Filter konfiguriert werden, der die gemessene Eingangsspannung an den Analogeingängen filtert. Der Filter verhält sich wie ein klassischer Tiefpassfilter, wobei die Grenzfrequenz durch den in die Systemvariable eingetragenen Wert eingestellt wird. Es sind Werte von 0...8 möglich.

Tabelle: Grenzfrequenz Software-Tiefpassfilter am Analogeingang

Ixx_FILTER	Filterfrequenz [Hz]	Signalanstiegszeit	Hinweise
0	Filter deaktiviert		
1	390	1 ms	
2	145	2,5 ms	
3	68	5 ms	
4	34	10 ms	empfohlen, Voreinstellung
5	17	21 ms	
6	8	42 ms	
7	4	84 ms	
8	2	169 ms	
≥ 9	34	10 ms	→ Voreinstellung

> ⓘ Nach dem Ändern der Filtereinstellung wird der Wert dieses Ein- oder Ausgangs nicht sofort richtig ausgegeben. Erst nach der Signalanstiegszeit (→ Tabelle) ist der Wert wieder korrekt.
>
> ⓘ Die Signalanstiegszeit ist die Zeitdauer, die ein Signal am Ausgang des Filters benötigt, um von 10 % auf 90 % des Endwerts zu kommen, wenn am Eingang ein Sprung angelegt wird. Die Signalabstiegszeit ist die Zeitdauer von 90 % bis 10 %.

Analogeingänge: Konfiguration und Diagnose

13960

- ► Die Konfiguration jedes einzelnen Eingangs erfolgt über das Anwendungsprogramm:
  - FB *INPUT_ANALOG* (→ Seite 129) > Eingang MODE oder:
  - FB *SET_INPUT_MODE* (→ Seite 132) > Eingang MODE

> Werden die Analogeingänge auf Strommessung konfiguriert, wird bei Überschreiten des Endwertes (21,7 mA) in den sicheren Spannungsmessbereich (0...32 V DC) geschaltet und das jeweilige Fehlerbit im Merkerbyte ERROR_CURRENT_Ix gesetzt. Sinkt der Wert wieder unter den Grenzwert, schaltet der Eingang selbsttätig auf den Strommessbereich zurück.

> Im Anwendungsprogramm können die Systemvariablen ANALOG00...ANALOGxx zur kundenspezifischen Diagnose der Eingänge dienen.

18414

> ! Falls Eingang I15 nicht verwendet:
> ► Eingang I15 als Binäreingang konfigurieren!

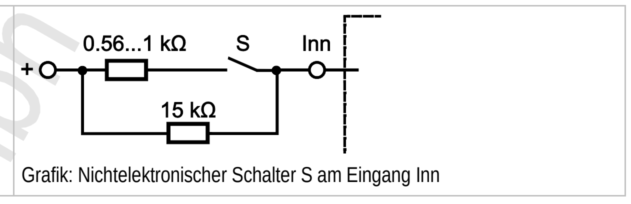
Binäreingänge: Konfiguration und Diagnose

14516

- ► Die Konfiguration jedes einzelnen Eingangs erfolgt über das Anwendungsprogramm:
  - FB *INPUT_ANALOG* (→ Seite 129) > Eingang MODE oder:
  - FB *SET_INPUT_MODE* (→ Seite 132) > Eingang MODE
- ► Für NAMUR: Soll die Diagnose genutzt werden, dann diesen Modus zusätzlich aktivieren:
  - FB SET_INPUT_MODE > Eingang DIAGNOSTICS setzen.

NAMUR-Diagnose für digitale Signale bei nichtelektronischen Schaltern:

► Schalter mit einer zusätzlichen Widerstandsbeschaltung versehen!



Grafik: Nichtelektronischer Schalter S am Eingang Inn

13956

> Das Diagnose-Ergebnis zeigen z.B. folgende Systemmerker:

Systemmerker (Symbolname)	Typ	Beschreibung
ERROR_BREAK_Ix (x=0...n; Wert abhängig vom Gerät, → Datenblatt)	DWORD	Eingangsgruppe x: Leiterbruch-Fehler oder (Widerstandseingang): Schluss nach Versorgung [Bit 0 für Eingang 0] ... [Bit z für Eingang z] dieser Gruppe Bit = TRUE: Fehler Bit = FALSE: kein Fehler
ERROR_SHORT_Ix (x=0...n; Wert abhängig vom Gerät, → Datenblatt)	DWORD	Eingangsgruppe x: Kurzschluss-Fehler [Bit 0 für Eingang 0] ... [Bit z für Eingang z] dieser Gruppe Bit = TRUE: Fehler Bit = FALSE: kein Fehler

> Im Anwendungsprogramm können die Systemvariablen ANALOG00...ANALOGxx zur kundenspezifischen Diagnose der Eingänge dienen.

Schnelle Eingänge

2193

Die Geräte verfügen über schnelle Zähl-/Impulseingänge für eine Eingangsfrequenz bis 30 kHz (→ Datenblatt).

Der Eingangswiderstand der schnellen Eingänge schaltet automatisch um, je nach verwendetem Modus oder Funktionsblock:

Eingangswiderstand	bei Modus / FB
3,2 kOhm	(Standard) FAST_COUNT, FREQUENCY, INC_ENCODER, PERIOD und ähnliche FBs
50,7 kOhm	Messeingang 32 V

Werden z.B. mechanische Schalter an diesen Eingängen angeschlossen, kann es durch Kontaktprellen zu Fehlsignalen in der Steuerung kommen.

▶ Bei Bedarf diese "Fehlsignale" über die Filter Ixx_DFILTER ausfiltern.
(→ Kapitel *Systemmerker* (→ Seite 221)) (nicht für alle Eingänge verfügbar)

Geeignete Funktionsbausteine sind z.B.:

FAST_COUNT (→ Seite 140)	Zählerbaustein für schnelle Eingangsimpulse
FREQUENCY (→ Seite 142)	misst die Frequenz des am gewählten Kanal ankommenden Signals
FREQUENCY_PERIOD (→ Seite 144)	misst die Frequenz und die Periodendauer (Zykluszeit) in [µs] am angegebenen Kanal
INC_ENCODER (→ Seite 146)	Vorwärts-/Rückwärts-Zählerfunktion zur Auswertung von Drehgebern
PERIOD (→ Seite 148)	misst am angegebenen Kanal die Frequenz und die Periodendauer (Zykluszeit) in [µs]
PERIOD_RATIO (→ Seite 150)	misst die Frequenz und die Periodendauer (Zykluszeit) in [µs] über die angegebenen Perioden am angegebenen Kanal. Zusätzlich wird das Puls-/Periodenverhältnis in [‰] angegeben.
PHASE (→ Seite 152)	liest ein Kanalpaar mit schnellen Eingängen ein und vergleicht die Phasenlage der Signale

Bei Einsatz dieser Bausteine werden automatisch die dort parametrierten Ein-/Ausgänge konfiguriert. Der Programmierer der Anwendung ist hiervon entlastet.

Hardware-Filter konfigurieren

9154

Über die Systemvariable Ixx_DFILTER kann ein digitaler Hardware-Filter an den schnellen Zähl- und Impulseingängen konfiguriert werden. Der Wert in µs (max. 100 000) gibt an, wie lange ein binärer Pegel ohne Unterbrechung anliegen muss, bevor er übernommen wird. Voreinstellung = 0 µs.

> ! Der Pegelwechsel des Eingangssignals wird um den im Filter eingestellten Wert verzögert.

Nur bei folgenden Funktionsbausteinen hat der Filter Auswirkungen auf die erfassten Signale:

FAST_COUNT (→ Seite 140)	Zählerbaustein für schnelle Eingangsimpulse
FREQUENCY (→ Seite 142)	misst die Frequenz des am gewählten Kanal ankommenden Signals
FREQUENCY_PERIOD (→ Seite 144)	misst die Frequenz und die Periodendauer (Zykluszeit) in [µs] am angegebenen Kanal
INC_ENCODER (→ Seite 146)	Vorwärts-/Rückwärts-Zählerfunktion zur Auswertung von Drehgebern
PERIOD (→ Seite 148)	misst am angegebenen Kanal die Frequenz und die Periodendauer (Zykluszeit) in [µs]
PERIOD_RATIO (→ Seite 150)	misst die Frequenz und die Periodendauer (Zykluszeit) in [µs] über die angegebenen Perioden am angegebenen Kanal. Zusätzlich wird das Puls-/Periodenverhältnis in [‰] angegeben.

Digitale Filter stehen nicht für alle schnellen Zähl- und Impulseingänge zur Verfügung.

Einsatz als Binäreingänge

3804

Durch die zulässigen hohen Eingangsfrequenzen können auch Fehlsignale erkannt werden, z.B. prellende Kontakte mechanischer Schalter.

► Bei Bedarf die Fehlsignale im Anwendungsprogramm unterdrücken!

4.4.2 Ausgänge konfigurieren

Inhalt

3976

Zulässige Betriebsarten → Kapitel *Mögliche Betriebsarten Ein-/Ausgänge* (→ Seite 232)

Software-Filter der Ausgänge konfigurieren

6882

Über die Systemvariablen Qxx_FILTER kann ein Software-Filter konfiguriert werden, der die gemessenen Stromwerte filtert. Der Filter verhält sich wie ein klassischer Tiefpassfilter, wobei die Grenzfrequenz durch den in die Systemvariable eingetragenen Wert eingestellt wird.

Tabelle: Grenzfrequenz Software-Tiefpassfilter bei der Strommessung am Ausgang

Qxx_FILTER	Filterfrequenz [Hz]	Signalanstiegszeit	Hinweise
0	Filter deaktiviert		
1	580	0,6 ms	
2	220	1,6 ms	
3	102	3,5 ms	
4	51	7 ms	empfohlen, Voreinstellung
5	25	14 ms	
6	12	28 ms	
7	6	56 ms	
8	3	112 ms	
≥ 9	51	7 ms	→ Voreinstellung

Nach dem Ändern der Filtereinstellung wird der Wert dieses Ein- oder Ausgangs nicht sofort richtig ausgegeben. Erst nach der Signalanstiegszeit (→ Tabelle) ist der Wert wieder korrekt.

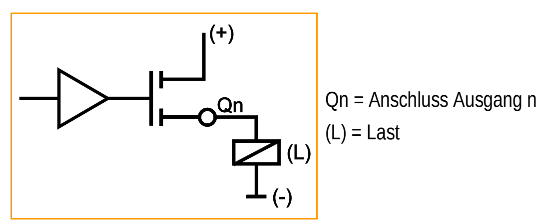
Die Signalanstiegszeit ist die Zeitdauer, die ein Signal am Ausgang des Filters benötigt, um von 10 % auf 90 % des Endwerts zu kommen, wenn am Eingang ein Sprung angelegt wird. Die Signalabstiegszeit ist die Zeitdauer von 90 % bis 10 %.

Binärausgänge: Konfiguration und Diagnose

15754

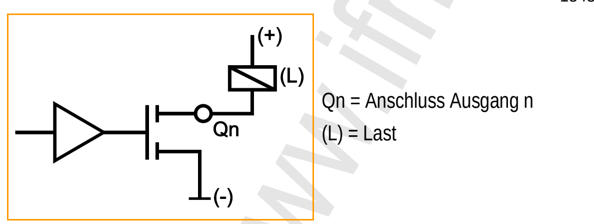
Bei den Geräte-Ausgängen sind folgende Betriebsarten möglich (→ Datenblatt):
• binärer Ausgang, plus-schaltend (BH) mit/ohne Diagnosefunktion
• binärer Ausgang, minus-schaltend (BL) ohne Diagnosefunktion

15450



Qn = Anschluss Ausgang n
(L) = Last

Prinzipschaltung Ausgang plus-schaltend (BH)
für positives Ausgangssignal



Qn = Anschluss Ausgang n
(L) = Last

Prinzipschaltung Ausgang minus-schaltend (BL)
für negatives Ausgangssignal

13975

⚠ WARNUNG

Gefährlicher Wiederanlauf möglich!
Gefahr von Personenschaden! Gefahr von Sachschaden an der Maschine/Anlage!

Wird ein Ausgang im Fehlerfall hardwaremäßig abgeschaltet, ändert sich der durch das Anwendungsprogramm erzeugte logische Zustand dadurch nicht.

► Abhilfe:
• Die Ausgänge zunächst im Anwendungsprogramm logisch zurücksetzen!
• Fehler beseitigen!
• Ausgänge situationsabhängig wieder setzen.

Binärausgänge: Konfiguration

15868

► Die Konfiguration jedes einzelnen Ausgangs erfolgt über das Anwendungsprogramm:
→ FB *SET_OUTPUT_MODE* (→ Seite 155) > Eingang MODE
zulässige Werte → Kapitel *Mögliche Betriebsarten Ein-/Ausgänge* (→ Seite 232)

Binärausgänge: Diagnose

15762

Soll die Diagnose genutzt werden, muss diese zusätzlich aktiviert werden.

► Ausgang als Binärausgang mit Diagnose nutzen (→ Datenblatt):
→ FB SET_OUTPUT_MODE > Eingang DIAGNOSTICS = TRUE

> Die Diagnosemeldungen der Ausgänge erscheinen im Systemmerker ERRORCODE:

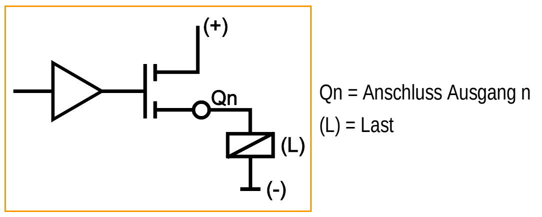
ERRORCODE	DWORD	Zuletzt eingetragener Fehler in der internen Fehlerliste Die Liste enthält alle aufgetretenen Fehler-Codes.

PWM-Ausgänge

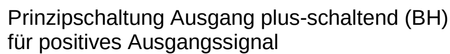
14717

Bei den Geräte-Ausgängen sind folgende Betriebsarten möglich (→ Datenblatt):
- PWM-Ausgang, plus-schaltend (BH) ohne Diagnosefunktion
- PWM-Ausgangspaar H-Brücke ohne Diagnosefunktion

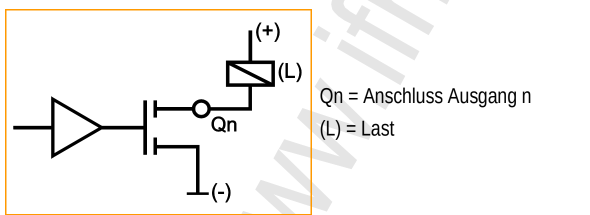
15450



Qn = Anschluss Ausgang n
(L) = Last



Prinzipschaltung Ausgang plus-schaltend (BH)
für positives Ausgangssignal



Qn = Anschluss Ausgang n
(L) = Last

Prinzipschaltung Ausgang minus-schaltend (BL)
für negatives Ausgangssignal

9980

HINWEIS

PWM-Ausgänge dürfen NICHT parallel betrieben werden, um z.B. den max. Ausgangsstrom zu erhöhen. Die Ausgänge arbeiten nicht synchron.

Andernfalls kann die komplette Last über nur einen Ausgang gehen. Die Strommessung funktioniert dann nicht mehr.

- PWM-Ausgänge können mit und ohne Stromregelfunktion betrieben werden.
  Stromgeregelte PWM-Ausgänge werden überwiegend zur Ansteuerung von proportionalen Hydraulikfunktionen genutzt.

Verfügbarkeit von PWM

15885

PWM-fähige Ausgänge → Datenblatt

FBs für PWM-Funktionen

14710

Für die PWM-Funktion der Ausgänge stehen folgende Funktionsbausteine zur Verfügung:

OUTPUT_BRIDGE (→ Seite 159)	H-Brücke an einem PWM-Kanalpaar
OUTPUT_CURRENT (→ Seite 163)	misst den Strom (Mittelung über Dither-Periode) an einem Ausgangskanal
OUTPUT_CURRENT_CONTROL (→ Seite 164)	Stromregler für einen PWMi-Ausgangskanal
PWM1000 (→ Seite 167)	initialisiert und parametriert einen PWM-fähigen Ausgangskanal das Puls-Pausen-Verhältnis kann in 1 ‰-Schritten angegeben werden

Mögliche Spezial-Funktionen der Ausgänge:
→ Kapitel *Bausteine: Hydraulikregelung* (→ Seite 169)
→ Kapitel *Bausteine: Regler* (→ Seite 184)

Stromregelung mit PWM (= PWMi)

13829

Über die im Controller integrierten Strommesskanäle kann eine Strommessung des Spulenstroms durchgeführt werden. Dadurch kann zum Beispiel der Strom bei einer Spulenerwärmung nachgeregelt werden. Damit bleiben die Hydraulikverhältnisse im System gleich.

Grundsätzlich sind die stromgeregelten Ausgänge gegen Kurzschluss geschützt.

4.5 Variablen

Inhalt

3130

In diesem Kapitel erfahren Sie mehr über den Umgang mit Variablen.

14486

Das Gerät unterstützt folgende Variablentypen:

Variable	Deklarationsort	Gültigkeitsbereich	Speicherverhalten
lokal	im Deklarationsteil des Bausteins	gilt nur im Baustein (POU), in dem sie konfiguriert wurde	flüchtig
lokal Retain			nicht flüchtig
global	in [Ressourcen] > [Globale Variablen] > [Globale_Variablen]	gilt in allen Bausteinen (POUs) dieses CODESYS-Projekts	flüchtig
global Retain			nicht flüchtig
Netzwerk	in [Ressourcen] > [Globale Variablen] > Deklarationsliste	Werte stehen allen CODESYS-Projekten im gesamten Netzwerk zur Verfügung, wenn die Variable in ihren Deklarationslisten enthalten ist.	flüchtig
Netzwerk Retain			nicht flüchtig



→ CODESYS-Programmierhandbuch
→ **ecomat*mobile***-DVD "Software, tools and documentation"

4.5.1 Retain-Variablen

15454

Als `RETAIN` deklarierte Variablen erzeugen remanente Daten. Retain-Variablen behalten beim Aus-/Einschalten des Geräts oder einem Online-Reset die in ihnen gespeicherten Werte.

> ❗ Die Inhalte der Retain-Variablen gehen verloren, falls beim Ausschalten das Gerät im STOP-Zustand ist!

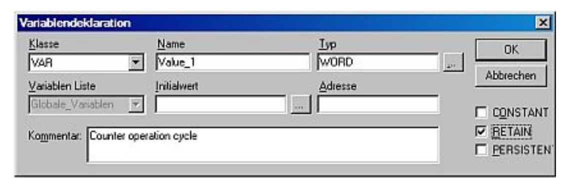
14166

Typische Einsätze für Retain-Variablen sind z.B.:
- Betriebsstunden, die zur Laufzeit der Maschine fortgeschrieben werden,
- Positionswerte von Inkrementalgebern,
- im Bildschirmgerät eingetragene Sollwerte,
- Maschinenparameter,

also alle Variablen, deren Werte beim Ausschalten des Geräts nicht verloren gehen dürfen.

Als Retain können alle Variablentypen, auch komplexe Stukturen (z.B. Timer), gekennzeichnet werden.

► Dazu in der Variablen-Deklaration das Kontrollfeld [RETAIN] aktivieren (→ Bild).



4.5.2 Netzwerkvariablen

9856

Globale Netzwerkvariablen dienen dem Datenaustausch zwischen Controllern im Netzwerk. Die Werte von globalen Netzwerkvariablen stehen allen CODESYS-Projekten im gesamten Netzwerk zur Verfügung, wenn die Variablen in deren Deklarationslisten enthalten sind.

► Dazu folgende Bibliothek(en) in das CODESYS-Projekt einbinden:

- `3S_CANopenNetVar.lib`

5 ifm-Funktionselemente

Inhalt

13586

Alle CODESYS-Funktionselemente (FBs, PRGs, FUNs) sind in Bibliotheken zusammengefasst. Nachfolgend zeigen wir Ihnen alle **ifm**-Bibliotheken, die Sie zusammen mit diesem Gerät nutzen können.

Anschließend finden Sie eine thematisch gegliederte Beschreibung der Funktionselemente.

5.1 ifm-Bibliotheken für das Gerät CR0033

Inhalt

14235

Legende für ..._Vxxyyzz.LIB:

V	Version
xx: 00...99	Versionsnummer
yy: 00...99	Release-Nummer
zz: 00...99	Patch-Nummer

Hier finden Sie die für dieses Gerät passenden **ifm**-Funktionselemente aufgelistet, nach CODESYS-Bibliotheken sortiert.

5.1.1 Bibliothek ifm_CR0033_V010009.LIB

15767

Dies ist die Geräte-Bibliothek. Diese **ifm**-Bibliothek enthält folgende Bausteine:

Baustein	Kurzbeschreibung
CANx (→ Seite 74)	initialisiert die CAN-Schnittstelle x x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)
CANx_BAUDRATE (→ Seite 75)	stellt die Übertragungsrate für den Busteilnehmer an der CAN-Schnittstelle x ein x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)
CANx_BUSLOAD (→ Seite 76)	ermittelt die aktuelle Buslast an der CAN-Schnittstelle x und zählt die aufgetretenen Error-Frames x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)
CANx_DOWNLOADID (→ Seite 78)	stellt den Download-Identifier für die CAN-Schnittstelle x ein x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)
CANx_ERRORHANDLER (→ Seite 79)	führt ein "manuelles" Bus-Recover auf der CAN-Schnittstelle x durch x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)
CANx_RECEIVE (→ Seite 80)	CAN-Schnittstelle x: konfiguriert ein Datenempfangsobjekt und liest den Empfangspuffer des Datenobjektes aus x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)
CANx_SDO_READ (→ Seite 102)	CAN-Schnittstelle x: liest das SDO mit den angegebenen Indizes aus dem Knoten aus x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)
CANx_SDO_WRITE (→ Seite 104)	CAN-Schnittstelle x: schreibt das SDO mit den angegebenen Indizes in den Knoten x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)
CANx_TRANSMIT (→ Seite 82)	übergibt in jedem Aufruf ein CAN-Datenobjekt (Message) an die CAN-Schnittstelle x zur Übertragung x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)
CHECK_DATA (→ Seite 212)	erzeugt über einen konfigurierbaren Speicherbereich eine Prüfsumme (CRC) und prüft die Daten des Speicherbereichs auf ungewollte Veränderung
DELAY (→ Seite 185)	verzögert die Ausgabe des Eingangswertes um die Zeit T (Totzeit-Glied)
FAST_COUNT (→ Seite 140)	Zählerbaustein für schnelle Eingangsimpulse
FLASHREAD (→ Seite 204)	liest unterschiedliche Datentypen direkt aus dem Flash-Speicher in den RAM
FLASHWRITE (→ Seite 205)	schreibt unterschiedliche Datentypen direkt in den Flash-Speicher
FRAMREAD (→ Seite 207)	liest unterschiedliche Datentypen direkt aus dem FRAM-Speicher in den RAM FRAM steht hier allgemein für alle Arten von nichtflüchtigen, schnellen Speichern.
FRAMWRITE (→ Seite 208)	schreibt unterschiedliche Datentypen direkt in den FRAM-Speicher FRAM steht hier allgemein für alle Arten von nichtflüchtigen, schnellen Speichern.
FREQUENCY (→ Seite 142)	misst die Frequenz des am gewählten Kanal ankommenden Signals
FREQUENCY_PERIOD (→ Seite 144)	misst die Frequenz und die Periodendauer (Zykluszeit) in [µs] am angegebenen Kanal
GET_IDENTITY (→ Seite 214)	liest die im Gerät gespeicherten spezifischen Kennungen: • Hardware-Name und Hardware-Version des Geräts • Name des Laufzeitsystems im Gerät • Version und Ausgabe des Laufzeitsystems im Gerät • Name der Anwendung (wurde zuvor mit *SET_IDENTITY* (→ Seite 216) gespeichert) • Seriennummer des Geräts
INC_ENCODER (→ Seite 146)	Vorwärts-/Rückwärts-Zählerfunktion zur Auswertung von Drehgebern
INPUT_ANALOG (→ Seite 129)	Strom- und Spannungsmessung am analogen Eingangskanal
MEMCPY (→ Seite 209)	schreibt und liest unterschiedliche Datentypen direkt in den Speicher
MEMORY_RETAIN_PARAM (→ Seite 201)	legt das remanente Verhalten der Daten für verschiedene Ereignisse fest
MEMSET (→ Seite 210)	beschreibt einen bestimmten Datenbereich
NORM (→ Seite 135)	normiert einen Wert [WORD] innerhalb festgelegter Grenzen auf einen Wert mit neuen Grenzen
NORM_DINT (→ Seite 137)	normiert einen Wert [DINT] innerhalb festgelegter Grenzen auf einen Wert mit neuen Grenzen
NORM_REAL (→ Seite 138)	normiert einen Wert [REAL] innerhalb festgelegter Grenzen auf einen Wert mit neuen Grenzen

Baustein	Kurzbeschreibung
OUTPUT_BRIDGE (→ Seite 159)	H-Brücke an einem PWM-Kanalpaar
OUTPUT_CURRENT (→ Seite 163)	misst den Strom (Mittelung über Dither-Periode) an einem Ausgangskanal
OUTPUT_CURRENT_CONTROL (→ Seite 164)	Stromregler für einen PWMi-Ausgangskanal
PERIOD (→ Seite 148)	misst am angegebenen Kanal die Frequenz und die Periodendauer (Zykluszeit) in [µs]
PERIOD_RATIO (→ Seite 150)	misst die Frequenz und die Periodendauer (Zykluszeit) in [µs] über die angegebenen Perioden am angegebenen Kanal. Zusätzlich wird das Puls-/Periodenverhältnis in [‰] angegeben.
PHASE (→ Seite 152)	liest ein Kanalpaar mit schnellen Eingängen ein und vergleicht die Phasenlage der Signale
PID1 (→ Seite 186)	PID-Regler
PID2 (→ Seite 188)	PID-Regler
PT1 (→ Seite 190)	Regelstrecke mit Verzögerung 1. Ordnung
PWM1000 (→ Seite 167)	initialisiert und parametriert einen PWM-fähigen Ausgangskanal das Puls-Pausen-Verhältnis kann in 1 ‰-Schritten angegeben werden
SERIAL_PENDING (→ Seite 119)	ermittelt die Anzahl der im seriellen Empfangspuffer gespeicherten Datenbytes
SERIAL_RX (→ Seite 120)	liest mit jedem Aufruf ein empfangenes Datenbyte aus dem seriellen Empfangspuffer aus
SERIAL_SETUP (→ Seite 121)	initialisiert die serielle RS232-Schnittstelle
SERIAL_TX (→ Seite 122)	überträgt ein Datenbyte über die serielle RS232-Schnittstelle
SET_DEBUG (→ Seite 215)	organisiert (abhängig vom TEST-Eingang) den DEBUG-Modus oder den Monitoring-Modus
SET_IDENTITY (→ Seite 216)	setzt eine anwendungsspezifische Programmkennung
SET_INPUT_MODE (→ Seite 132)	weist einem Eingangskanal eine Betriebsart zu
SET_INTERRUPT_I (→ Seite 124)	bedingtes Ausführen eines Programmteils nach einer Interrupt-Anforderung über einen definierten Eingangskanal
SET_INTERRUPT_XMS (→ Seite 126)	bedingtes Ausführen eines Programmteils im Intervall von x Millisekunden
SET_OUTPUT_MODE (→ Seite 155)	setzt die Betriebsart des gewählten Ausgangskanals
SET_PASSWORD (→ Seite 217)	setzt Benutzerkennung für Zugangskontrolle bei Programm- und Speicher-Upload
SOFTRESET (→ Seite 192)	führt einen kompletten Neustart des Geräts aus
TEMPERATURE (→ Seite 197)	liest die aktuelle Temperatur im Gerät aus
TIMER_READ (→ Seite 194)	liest die aktuelle Systemzeit in [ms] aus Max-Wert = 49d 17h 2min 47s 295ms
TIMER_READ_US (→ Seite 195)	liest die aktuelle Systemzeit in [µs] aus Max-Wert = 1h 11min 34s 967ms 295µs

5.1.2 Bibliothek ifm_CR0033_CANopenxMaster_Vxxyyzz.LIB

13707

x = 1...4 = Nummer der CAN-Schnittstelle
Diese Bibliothek enthält Bausteine für den Betrieb des Geräts als CANopen-Master.

Diese **ifm**-Bibliothek enthält folgende Bausteine:

Baustein	Kurzbeschreibung
CANx_MASTER_EMCY_HANDLER (→ Seite 84)	verwaltet den geräteeigenen Fehlerstatus des CANopen-Masters an der CAN-Schnittstelle x x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)
CANx_MASTER_SEND_EMERGENCY (→ Seite 85)	versendet anwendungsspezifische Fehlerstatus des CANopen-Masters an der CAN-Schnittstelle x x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)
CANx_MASTER_STATUS (→ Seite 87)	Status-Anzeige an der CAN-Schnittstelle x des als CANopen-Master eingesetzten Gerätes x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

5.1.3 Bibliothek ifm_CR0033_CANopenxSlave_Vxxyyzz.LIB

13709

x = 1...4 = Nummer der CAN-Schnittstelle
Diese Bibliothek enthält Bausteine für den Betrieb des Geräts als CANopen-Slave.

Diese **ifm**-Bibliothek enthält folgende Bausteine:

Baustein	Kurzbeschreibung
CANx_SLAVE_EMCY_HANDLER (→ Seite 94)	verwaltet den geräteeigenen Fehlerstatus des CANopen-Slaves an der CAN-Schnittstelle x: • Error Register (Index 0x1001) und • Error Field (Index 0x1003) des CANopen Objektverzeichnis x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)
CANx_SLAVE_NODEID (→ Seite 95)	ermöglicht das Einstellen der Node-ID eines CANopen-Slaves an der CAN-Schnittstelle x zur Laufzeit des Anwendungsprogramms x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)
CANx_SLAVE_SEND_EMERGENCY (→ Seite 96)	versendet anwendungsspezifische Fehlerstatus des CANopen-Slaves an der CAN-Schnittstelle x x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)
CANx_SLAVE_SET_PREOP (→ Seite 98)	schaltet den Betriebsmodus dieses CANopen-Slaves an der CAN-Schnittstelle x von OPERATIONAL auf PRE-OPERATIONAL x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)
CANx_SLAVE_STATUS (→ Seite 99)	zeigt den Status des an der CAN-Schnittstelle x als CANopen-Slave eingesetzten Gerätes x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

5.1.4 Bibliothek ifm_CR0033_J1939_Vxxyyzz.LIB

13711

Diese Bibliothek enthält Bausteine zur Motorsteuerung.

Diese **ifm**-Bibliothek enthält folgende Bausteine:

Baustein	Kurzbeschreibung
J1939_x (→ Seite 107)	CAN-Schnittstelle x: Protokoll-Handler für das Kommunikationsprofil SAE J1939 x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)
J1939_x_GLOBAL_REQUEST (→ Seite 108)	CAN-Schnittstelle x: organisiert globales Anfordern und Empfangen von Daten der J1939-Netzwerkteilnehmer x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)
J1939_x_RECEIVE (→ Seite 110)	CAN-Schnittstelle x: empfängt eine einzelne Nachricht oder einen Nachrichtenblock x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)
J1939_x_RESPONSE (→ Seite 112)	CAN-Schnittstelle x: organisiert die automatische Antwort auf ein Request-Telegramm x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)
J1939_x_SPECIFIC_REQUEST (→ Seite 114)	CAN-Schnittstelle x: automatisches Anfordern einzelner Nachrichten von einem bestimmten (specific) J1939-Netzwerkteilnehmer x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)
J1939_x_TRANSMIT (→ Seite 116)	CAN-Schnittstelle x: versendet einzelne Nachrichten oder Nachrichtenblocks x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

5.1.5 Bibliothek ifm_hydraulic_32bit_Vxxyyzz.LIB

13729

Diese Bibliothek enthält Bausteine für Hydraulik-Steuerungen.

Diese **ifm**-Bibliothek enthält folgende Bausteine:

Baustein	Kurzbeschreibung
CONTROL_OCC (→ Seite 170)	OCC = Output Current Control (= stromgeregelter Ausgang) skaliert den Eingangswert [WORD] auf einen angegebenen Strombereich
JOYSTICK_0 (→ Seite 172)	skaliert Signale [INT] aus einem Joystick auf fest definierte Kennlinien, normiert auf 0…1000
JOYSTICK_1 (→ Seite 175)	skaliert Signale [INT] aus einem Joystick auf parametrierbare Kennlinien, normiert auf 0…1000
JOYSTICK_2 (→ Seite 179)	skaliert Signale [INT] aus einem Joystick auf einen parametrierbaren Kennlinien-Verlauf; die Normierung ist frei bestimmbar
NORM_HYDRAULIC (→ Seite 182)	normiert einen Wert [DINT] innerhalb festgelegter Grenzen auf einen Wert mit neuen Grenzen

5.2 ifm-Bausteine für das Gerät CR0033

Inhalt

13988
3826

Hier finden Sie die Beschreibung der für dieses Gerät passenden **ifm**-Funktionselemente, nach Thema sortiert.

5.2.1 Bausteine: CAN Layer 2

Inhalt

13754

Hier werden die CAN-Funktionsbausteine (Layer 2) zur Nutzung im Anwendungsprogramm beschrieben.

CANx

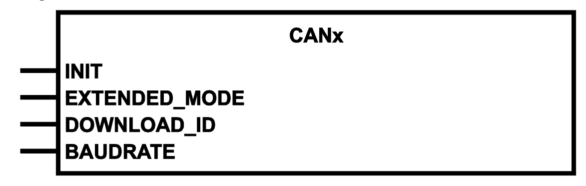
2159

x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

2162

CANx initialisiert die x. CAN-Schnittstelle.
(x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt))

Der Download-ID muss für jede Schnittstelle unterschiedlich sein.

Die Baudraten der einzelnen CANx können unterschiedlich eingestellt werden.

► Den Eingang INIT nur für einen Zyklus bei Neustart oder Restart der Schnittstelle setzen!

> ! Eine Änderung des Download-ID und/oder der Baudrate wird erst gültig ...
> • nach Spannung Aus/Ein,
> • nach Soft-Reset.

Wenn der FB nicht ausgeführt wird, arbeitet die Schnittstelle mit 11-Bit-Identifier.

Parameter der Eingänge

2163

Parameter	Datentyp	Beschreibung
INIT	BOOL	TRUE (im 1. Zyklus): Baustein wird initialisiert FALSE: im weiteren Programmablauf
EXTENDED_MODE	BOOL := FALSE	TRUE: Identifier der CAN-Schnittstelle arbeitet mit 29 Bits FALSE: Identifier der CAN-Schnittstelle arbeitet mit 11 Bits
DOWNLOAD_ID	BYTE	Download-ID der CAN-Schnittstelle x x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt) zulässig = 1...127 voreingestellt = 127 - (x-1)
BAUDRATE	WORD := 125	Baudrate [kBit/s] zulässig = 20, 50, 100, 125, 250, 500, 1000

CANx_BAUDRATE

11834

x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

11839

CANx_BAUDRATE stellt die Übertragungsrate für den Busteilnehmer ein.

Mit dem FB wird für das Gerät die Übertragungsrate eingestellt. Dazu wird am Eingang BAUDRATE der entsprechende Wert in kBit/s angegeben.

> ! Der neue Wert wird erst nach einem RESET gültig (Spannung Aus/Ein oder Soft-Reset).

Parameter der Eingänge

655

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE (im 1. Zyklus): Parameter übernehmen und aktivieren sonst: diese Funktion wird nicht ausgeführt
BAUDRATE	WORD := 125	Baudrate [kBit/s] zulässig = 20, 50, 100, 125, 250, 500, 1000

CANx_BUSLOAD

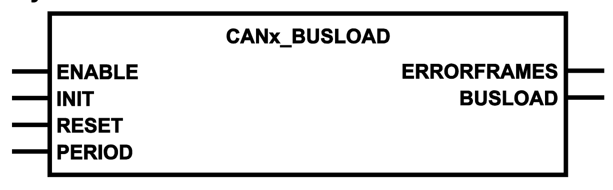
2178

x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

2180

Ermittelt die aktuelle Buslast auf dem CAN-Bus und zählt die aufgetretenen Error-Frames.

CANx_BUSLOAD ermittelt die Buslast über die Anzahl und Länge der während der Zeit PERIOD über den CAN-Bus übertragenen Telegramme, bei Berücksichtigung der aktuellen Baudrate. Der Wert BUSLOAD wird jeweils nach Ablauf der Zeit PERIOD aktualisiert.

Ist das Bit RESET dauerhaft FALSE, wird die Anzahl der Error-Frames angezeigt, die seit dem letzten RESET aufgetreten sind.

HINWEIS

Läuft die Kommunikation auf dem CAN-Bus über das CANopen-Protokoll, dann ist es sinnvoll, den Wert von PERIOD auf die Dauer des SYNC-Zyklus zu setzen.

Die Messperiode ist dabei nicht mit dem CANopen SYNC-Zyklus synchronisiert.

Parameter der Eingänge

2181

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
INIT	BOOL	TRUE (nur 1 Zyklus lang): Konfiguration der Messdauer PERIOD FALSE: im weiteren Programmablauf
RESET	BOOL	TRUE: ERRORFRAME zurücksetzen auf "0" FALSE: Funktion wird nicht ausgeführt
PERIOD	WORD	Zeit in [ms], über welche die Buslast ermittelt wird zulässig = 20...1 000 ms

Parameter der Ausgänge

2182

Parameter	Datentyp	Beschreibung
ERRORFRAMES	WORD	Anzahl der auf dem CAN-Bus aufgetretenen Error-Frames seit dem letzten Reset
BUSLOAD	BYTE	mittlere Buslast in [%] zulässig: 0...100

CANx_DOWNLOADID

11841

= CANx Download-ID

x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

11846

CANx_DOWNLOADID stellt den Download-Identifier für die CAN-Schnittstelle x ein.

Mit dem FB kann der Kommunikations-Identifier für den Programm-Download und das Debuggen eingestellt werden. Der neue Wert wird eingetragen, wenn der Eingang ENABLE auf TRUE gesetzt wird.

Der neue Wert wird erst nach einem RESET gültig (Spannung Aus/Ein oder Soft-Reset).

Parameter der Eingänge

649

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE (im 1. Zyklus): Parameter übernehmen und aktivieren sonst: diese Funktion wird nicht ausgeführt
ID	BYTE	Download-ID der CAN-Schnittstelle x setzen x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt) zulässig = 1...127 voreingestellt = 127 - (x-1)

CANx_ERRORHANDLER

2174

x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:

```
        CANx_ERRORHANDLER
—BUSOFF_RECOVER
```

Beschreibung

2329
13991

> ! Wenn die automatische Bus-Recover-Funktion genutzt werden soll (Voreinstellung), darf CANx_ERRORHANDLER **nicht** in das Programm eingebunden und instanziert werden!

CANx_ERRORHANDLER führt ein "manuelles" Bus-Recover auf der CAN-Schnittstelle x durch.

► Nach einem erkannten CAN-Busoff den FB für einen Zyklus mit BUSOFF_RECOVER = TRUE aufrufen, damit die Steuerung wieder auf dem CAN-Bus senden und empfangen kann.

► Anschließend im Anwendungsprogramm für diese CAN-Schnittstelle das Fehlerbit CANx_BUSOFF zurücksetzen.

> Die CAN-Schnittstelle arbeitet wieder.

Parameter der Eingänge

2177

Parameter	Datentyp	Beschreibung
BUSOFF_RECOVER	BOOL	TRUE (nur 1 Zyklus lang): > Bus-off-Zustand beheben > Neustart der CAN-Schnittstelle FALSE: Funktion wird nicht ausgeführt

CANx_RECEIVE

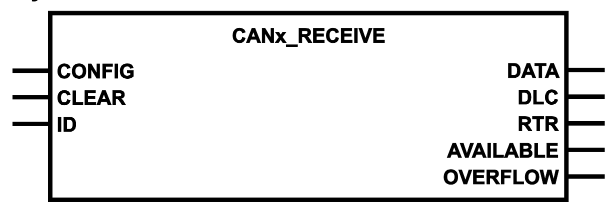
627

x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

13338

CANx_RECEIVE konfiguriert ein Datenempfangsobjekt und liest den Empfangspuffer des Datenobjektes aus.

► Den FB für jedes Datenobjekt in der Initialisierungsphase einmalig aufrufen, um dem CAN-Controller die Identifier der Datenobjekte bekannt zu machen.

► Im weiteren Programmzyklus CANx_RECEIVE zum Auslesen des jeweiligen Empfangspuffers aufrufen, bei langen Programmzyklen auch mehrfach.

► Im FB CANx einstellen, ob CANx_RECEIVE Normal oder Extended Frames empfangen soll.

> Wird CANx_RECEIVE für den Empfang eines Normal Frame konfiguriert, wird der Frame mit dieser ID nicht mehr an einen eventuell vorhandenen CANopen Stack weitergeleitet.

> Wird eine ID außerhalb des zulässigen Bereichs eingestellt (abhängig von der Einstellung in CANx), wird der Funktionsbaustein nicht ausgeführt.

► Den Ausgang AVAILABLE auswerten, so dass neu eingegangene Datenobjekte rechtzeitig aus dem Puffer gelesen und weiterverarbeitet werden.
Receive-Puffer: max. 16 Software-Puffer pro Identifier.

> Jeder Aufruf des FB dekrementiert das Byte AVAILABLE um 1.
Ist AVAILABLE = 0, sind keine Daten im Puffer.

► Nur bei Extended Frame:
Den Ausgang OVERFLOW auswerten, um einen Überlauf des Datenpuffers zu erkennen.
Wenn OVERFLOW = TRUE, dann ist mindestens 1 Datenobjekt verloren gegangen.
(Beim Standard Frame ist der CANopen-Stack nicht betroffen.)

Parameter der Eingänge

2172

Parameter	Datentyp	Beschreibung
CONFIG	BOOL	TRUE (im 1. Zyklus): Datenobjekt konfigurieren FALSE: im weiteren Programmablauf
CLEAR	BOOL	TRUE: Empfangspuffer löschen FALSE: Funktion wird nicht ausgeführt
ID	DWORD	Nummer des Datenobjekt-Identifiers: Normal Frame (2^{11} IDs): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2^{29} IDs): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF

Parameter der Ausgänge

19810

Parameter	Datentyp	Beschreibung
DATA	ARRAY [0..7] OF BYTE	empfangene Daten (1...8 Bytes)
DLC	BYTE	Anzahl der mit RDO empfangenen Bytes im Array DATA zulässig: 0...8
RTR	BOOL = FALSE	empfangene Nachricht war ein Remote Transmission Request (wird hier nicht unterstützt)
AVAILABLE	BYTE	Anzahl der verbleibenden Datenbytes zulässig = 0...16 0 = keine gültigen Daten vorhanden
OVERFLOW	BOOL	TRUE: Überlauf des Datenpuffers ⇨ Datenverlust! FALSE: Datenpuffer ist ohne Datenverlust

CANx_TRANSMIT

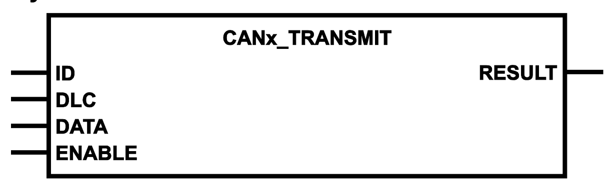
609

x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

2166

CANx_TRANSMIT übergibt in jedem Aufruf ein CAN-Datenobjekt (Message) an den CAN-Controller zur Übertragung.

► Den FB für jedes Datenobjekt im Programmzyklus aufgerufen, bei langen Programmzyklen auch mehrfach.
Transmit-Puffer: max. 16 Software- und 1 Hardware-Puffer für alle Identifier zusammen.

► Den Ausgang RESULT auswerten zur Prüfung, dass der Sendeauftrag angenommen wurde.

ℹ Vereinfacht gilt bei 125 kBit/s, dass pro 1 ms ein Sendeauftrag ausgeführt werden kann.

Über den Eingang ENABLE kann die Ausführung des FB zeitweilig gesperrt werden (ENABLE = FALSE). Damit kann z.B. eine Busüberlastung verhindert werden.

Mehrere Datenobjekte können quasi gleichzeitig verschickt werden, wenn jedem Datenobjekt ein Merkerflag zugeordnet wird und mit diesem die Ausführung des FB über den ENABLE-Eingang gesteuert wird.

Parameter der Eingänge

19813

Parameter	Datentyp	Beschreibung
ID	DWORD	Nummer des Datenobjekt-Identifiers: Normal Frame (2^{11} IDs): 0...2 047 = 0x0000 0000...0x0000 07FF Extended Frame (2^{29} IDs): 0...536 870 911 = 0x0000 0000...0x1FFF FFFF
DLC	BYTE	Anzahl der mit RDO zu übertragenden Bytes aus dem Array DATA zulässig: 0...8
DATA	ARRAY [0..7] OF BYTE	zu sendende Daten (1...8 Bytes)
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert

Parameter der Ausgänge

2168

Parameter	Datentyp	Beschreibung
RESULT	BOOL	TRUE (nur 1 Zyklus lang): der Baustein hat den Sendeauftrag angenommen FALSE: Sendeauftrag wurde nicht angenommen

5.2.2 Bausteine: CANopen-Master

Inhalt

1870

Für den CANopen-Master stellt **ifm electronic** eine Reihe von Bausteinen zur Verfügung, die im Folgenden erklärt werden.

CANx_MASTER_EMCY_HANDLER

2006

x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_CANopenxMaster_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

2009

CANx_MASTER_EMCY_HANDLER verwaltet den geräteeigenen Fehlerstatus des Masters. Der FB muss in folgenden Fällen aufgerufen werden:

- der Fehlerstatus soll ins Netzwerk übertragen werden und
- die Fehlermeldungen des Anwendungsprogramms sollen im Objektverzeichnis gespeichert werden.

Über den FB können die aktuellen Werte aus dem Error-Register (Index 0x1001/01) und Error Field (Index 0x1003/0-5) des CANopen-Objektverzeichnis ausgelesen werden.

> ! Sollen anwendungsspezifische Fehlernachrichten im Objektverzeichnis gespeichert werden, muss CANx_MASTER_EMCY_HANDLER **nach** dem (mehrfachen) Bearbeiten von *CANx_MASTER_SEND_EMERGENCY* (→ Seite 85) aufgerufen werden.

Parameter der Eingänge

2010

Parameter	Datentyp	Beschreibung
CLEAR_ERROR_FIELD	BOOL	FALSE ⇨ TRUE (Flanke): • Inhalt des ERROR_FIELD an FB-Ausgang ausgeben • Inhalt des ERROR_FIELD im Objektverzeichnis löschen sonst: diese Funktion wird nicht ausgeführt

Parameter der Ausgänge

2011

Parameter	Datentyp	Beschreibung
ERROR_REGISTER	BYTE	Zeigt den Inhalt des OBV Index 0x1001 (Error-Register)
ERROR_FIELD	ARRAY [0..5] OF WORD	Zeigt den Inhalt des OBV Index 0x1003 (Error-Field) ERROR_FIELD[0]: Anzahl der gespeicherten Fehler ERROR_FIELD[1...5]: gespeicherte Fehler, der jüngste Fehler steht im Index [1]

CANx_MASTER_SEND_EMERGENCY

2012

x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_CANopenxMaster_Vxxyyzz.LIB`

Symbol in CODESYS:

```
CANx_MASTER_SEND_EMERGENCY
ENABLE
ERROR
ERROR_CODE
ERROR_REGISTER
MANUFACTURER_ERROR_FIELD
```

Beschreibung

2015

CANx_MASTER_SEND_EMERGENCY versendet anwendungsspezifische Fehlerstatus. Der FB wird aufgerufen, wenn der Fehlerstatus an andere Geräte im Netzwerkverbund übertragen werden soll.

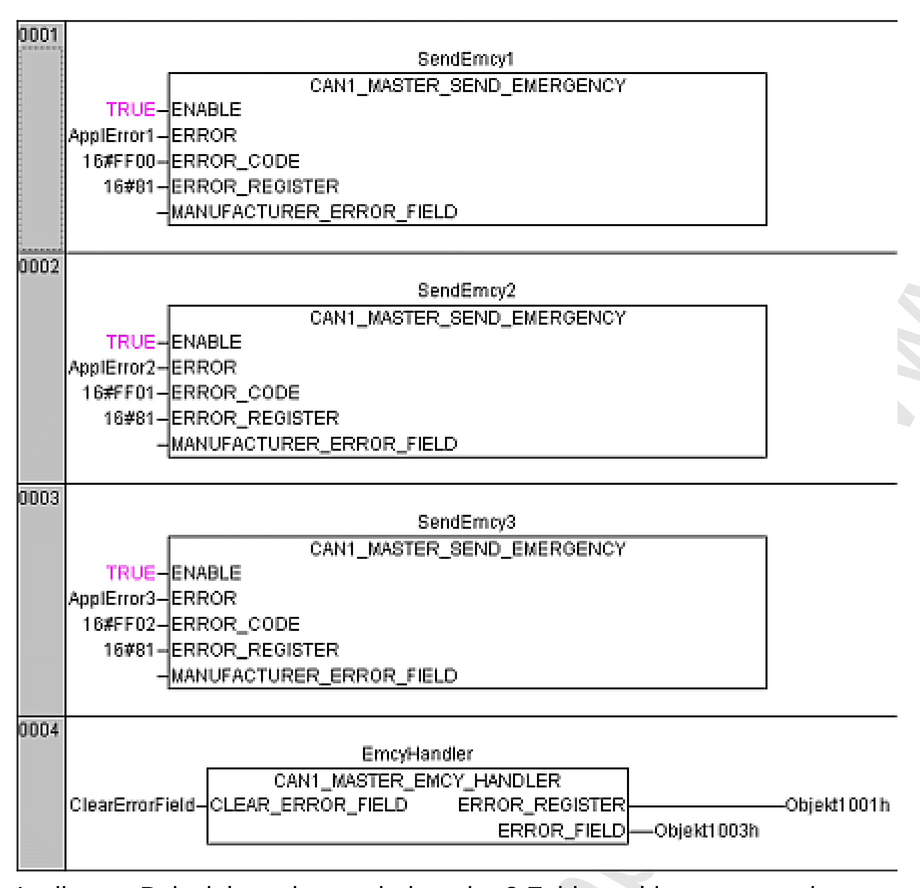
> ! Sollen anwendungsspezifische Fehlernachrichten im Objektverzeichnis gespeichert werden, muss *CANx_MASTER_EMCY_HANDLER* (→ Seite 84) **nach** dem (mehrfachen) Bearbeiten von CANx_MASTER_SEND_EMERGENCY aufgerufen werden.

Parameter der Eingänge

2016

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
ERROR	BOOL	Über diesen Eingang wird dem FB die Information übergeben, ob der zum konfigurierten Fehlercode gehörende Fehler aktuell anliegt. FALSE ⇨ TRUE (Flanke): sendet den anstehenden Fehler-Code falls Eingang in der letzten Sekunde nicht TRUE war TRUE ⇨ FALSE (Flanke) UND Fehler steht nicht mehr an: Nach Verzögerung von ca. 1 s: > Null-Fehlermeldung wird gesendet sonst: diese Funktion wird nicht ausgeführt
ERROR_CODE	WORD	Der Error-Code gibt detailliert Auskunft über den erkannten Fehler. Die Werte sollten gemäß der CANopen-Spezifikation eingetragen werden.
ERROR_REGISTER	BYTE	ERROR_REGISTER gibt die Art des Fehlers an. Der hier angegebene Wert wird mit allen anderen aktuell aktiven Fehlernachrichten bitweise ODER-verknüpft. Der sich hierbei ergebende Wert wird ins Error-Register (Index 1001_{16}/00) geschrieben und mit der EMCY-Nachricht versendet. Die Werte sollten gemäß der CANopen-Spezifikation eingetragen werden.
MANUFACTURER_ERROR_FIELD	ARRAY [0..4] OF BYTE	Hier können bis zu 5 Bytes anwendungsspezifische Fehlerinformationen eingetragen werden. Das Format ist dabei frei wählbar.

Beispiel: CANx_MASTER_SEND_EMERGENCY



In diesem Beispiel werden nacheinander 3 Fehlermeldungen generiert:

1. ApplError1, Code = 0xFF00 im Fehlerregister 0x81
2. ApplError2, Code = 0xFF01 im Fehlerregister 0x81
3. ApplError3, Code = 0xFF02 im Fehlerregister 0x81

Der FB CAN1_MASTER_EMCY_HANDLER sendet die Fehlermeldungen an das Fehler-Register "Objekt 0x1001" im Fehler-Array "Objekt 0x1003".

CANx_MASTER_STATUS

2692

x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_CANopenxMaster_Vxxyyzz.LIB`

Symbol in CODESYS:

CANx_MASTER_STATUS	
GLOBAL_START	NODE_ID
CLEAR_RX_OVERFLOW_FLAG	BAUDRATE
CLEAR_RX_BUFFER	NODE_STATE
CLEAR_TX_OVERFLOW_FLAG	SYNC
CLEAR_TX_BUFFER	RX_OVERFLOW
CLEAR_OD_CHANGED_FLAG	TX_OVERFLOW
CLEAR_ERROR_CONTROL	OD_CHANGED
RESET_ALL_NODES	ERROR_CONTROL
START_ALL_NODES	GET_EMERGENCY
NODE_STATE_SLAVE	FIRST_NODE_INDEX
EMERGENCY_OBJECT_SLAVES	LAST_NODE_INDEX

Beschreibung

2024

Status-Anzeige des als CANopen-Master eingesetzten Gerätes

Der FB zeigt den Status des als CANopen-Master eingesetzten Gerätes an. Weitere Möglichkeiten:
- den Status des Netzwerks überwachen
- den Status der angeschlossenen Slaves überwachen
- die Slaves im Netzwerk zurücksetzen oder starten.

Der FB vereinfacht die Anwendung der CODESYS-CANopen-Master-Bibliotheken. Wir empfehlen dringend, die Auswertung des Netzwerkstatus und der Fehlermeldungen über diesen FB vorzunehmen.

Parameter der Eingänge

19891

Parameter	Datentyp	Beschreibung
GLOBAL_START	BOOL	TRUE: Alle angeschlossenen Netzwerkteilnehmer (Slaves) werden gleichzeitig bei der Netzwerkinitialisierung gestartet (⇨ Zustand OPERATIONAL). FALSE: Die angeschlossenen Netzwerkteilnehmer werden einzeln nacheinander gestartet.
CLEAR_RX_OVERFLOW_FLAG	BOOL	FALSE ⇨ TRUE (Flanke): Fehlerflag RX_OVERFLOW löschen sonst: diese Funktion wird nicht ausgeführt
CLEAR_RX_BUFFER	BOOL	FALSE ⇨ TRUE (Flanke): Daten im Empfangspuffer löschen sonst: diese Funktion wird nicht ausgeführt
CLEAR_TX_OVERFLOW_FLAG	BOOL	FALSE ⇨ TRUE (Flanke): Fehlerflag TX_OVERFLOW löschen sonst: diese Funktion wird nicht ausgeführt
CLEAR_TX_BUFFER	BOOL	FALSE ⇨ TRUE (Flanke): Daten im Sendepuffer löschen sonst: diese Funktion wird nicht ausgeführt
CLEAR_OD_CHANGED_FLAG	BOOL	FALSE ⇨ TRUE (Flanke): Flag OD_CHANGED löschen sonst: diese Funktion wird nicht ausgeführt
CLEAR_ERROR_CONTROL	BOOL	FALSE ⇨ TRUE (Flanke): Die Guard-Fehlerliste (ERROR_CONTROL) löschen sonst: diese Funktion wird nicht ausgeführt
RESET_ALL_NODES	BOOL	FALSE ⇨ TRUE (Flanke): Alle angeschlossenen Netzwerkteilnehmer (Slaves) werden per NMT-Kommando zurückgesetzt sonst: diese Funktion wird nicht ausgeführt
START_ALL_NODES	BOOL	FALSE ⇨ TRUE (Flanke): Alle angeschlossenen Netzwerkteilnehmer (Slaves) werden per NMT-Kommando gestartet sonst: diese Funktion wird nicht ausgeführt
NODE_STATE_SLAVES	Array [0.. MAX_NODEINDEX] of CANx_NODE_STATE	In das Array werden die Statusinformationen der im CANopen-Netzwerk befindlichen Slaves geschrieben. Über den Zugriff auf bestimmte Werte in den Strukturen im Array kann auch das Verhalten der Slaves gesteuert werden. MAX_NODEINDEX ist eine Konstante, die beim Übersetzen der Anwendung von CODESYS ermittelt wird. Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben! Beispiel-Code → Kapitel *Beispiel: CANx_MASTER_STATUS* (→ Seite 90)
EMERGENCY_OBJECT_SLAVES	Array [0.. MAX_NODEINDEX] of CANx_EMERGENCY_MESSAGE	Zeigt die zuletzt aufgetretenen Fehlermeldungen aller Netzwerkknoten. Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben! → Kapitel *Zugriff auf die Strukturen zur Laufzeit der Anwendung* (→ Seite 92)

Parameter der Ausgänge

2696

Parameter	Datentyp	Beschreibung
NODE_ID	BYTE	aktuelle Knoten-ID des CANopen-Masters
BAUDRATE	WORD	aktuelle Baudrate des CANopen-Masters in [kBaud]
NODE_STATE	INT	aktueller Status des CANopen-Masters
SYNC	BOOL	SYNC-Signal des CANopen-Masters TRUE: Im letzten Zyklus wurde ein SYNC-Signal gesendet FALSE: Im letzten Zyklus wurde kein SYNC-Signal gesendet
RX_OVERFLOW	BOOL	TRUE: Fehler: Empfangspuffer-Überlauf FALSE: kein Überlauf
TX_OVERFLOW	BOOL	TRUE: Fehler: Sendepuffer-Überlauf FALSE: kein Überlauf
OD_CHANGED	BOOL	TRUE: Daten im Objektverzeichnis des CANopen-Masters wurden geändert FALSE: keine Datenänderung
ERROR_CONTROL	ARRAY [0..7] OF BYTE	Das Array enthält die Liste (max. 8) der fehlenden Netzwerkknoten (Guard- oder Heartbeat-Fehler) → Kapitel *Zugriff auf die Strukturen zur Laufzeit der Anwendung* (→ Seite 92)
GET_EMERGENCY	STRUCT CANx_EMERGENY_MESSAGE	Am Ausgang stehen die Daten für die Struktur CANx_EMERGENY_MESSAGE zur Verfügung. Es wird immer die zuletzt empfangene EMCY-Nachricht im CANopen-Netzwerk angezeigt. Um eine Liste aller aufgetretenen Fehler zu erhalten, muss das Array "EMERGENCY_OBJECT_SLAVES" ausgewertet werden.
FIRST_NODE_INDEX	INT	Bereich, in dem sich die Knotennummern der an diesem CAN-Bus angeschlossenen Knoten (Slaves) befinden
LAST_NODE_INDEX	INT	

Parameter der internen Strukturen

2698

Hier sehen Sie die Strukturen der in diesem Baustein genutzten Arrays.

Die Anwendung des FB CANx_MASTER_STATUS zeigen Ihnen die Code-Fragmente am Beispiel des Controllers CR0032 → Kapitel *Beispiel: CANx_MASTER_STATUS* (→ Seite 90).

Struktur von CANx_EMERGENCY_MESSAGE

13996

Die Struktur ist in den globalen Variablen der Bibliothek `ifm_CR0033_CANopenMaster_Vxxyyzz.LIB` angelegt.

Parameter	Datentyp	Beschreibung
NODE_ID	BYTE	Node-ID des Teilnehmers, von dem die EMCY-Nachricht empfangen wurde
ERROR_CODE	WORD	Error-Code mit der Information, welcher Fehler aufgetreten ist. → CANopen-Spezifikation CiA Draft Standard 301 Version 4
ERROR_REGISTER	BYTE	Wert im Error-Register (Index 0x1001/00) des sendenden Teilnehmers
MANUFACTURER_ERROR_FIELD	ARRAY [0..4] OF BYTE	herstellerspezifischer Datenbereich in der EMCY-Nachricht

Struktur von CANx_NODE_STATE

13997

Die Struktur ist in den globalen Variablen der Bibliothek `ifm_CR0033_CANopenMaster_Vxxyyzz.LIB` angelegt.

Parameter	Datentyp	Beschreibung
NODE_ID	BYTE	Node-ID des CANopen-Slaves, zu dem die Statusinformationen und Konfigurationsflags in der Struktur gehören
NODE_STATE	BYTE	aktueller Status des CANopen-Slaves aus Sicht des CANopen-Stacks des CANopen-Masters
LAST_STATE	BYTE	der letzte bekannte Status des CANopen-Slaves 0 = Bootup-Nachricht vom CANopen-Slave empfangen 4 = CANopen-Slave im Status PRE-OPERATIONAL und wird per SDO-Zugriff konfiguriert 5 = CANopen-Slave im Status OPERATIONAL 127 = CANopen-Slave im Status PRE-OPERATIONAL
RESET_NODE	BOOL	Flag zum manuellen Zurücksetzen des CANopen-Slaves (NMT-Kommando = `Reset_Node`)
START_NODE	BOOL	Flag zum manuellen Starten des CANopen-Slaves (NMT-Kommando = `start`)
PREOP_NODE	BOOL	Flag zum manuellen Versetzen des CANopen-Slaves in den Zustand PRE-OPERATIONAL (NMT-Kommando = `enter PRE-OPERATIONAL`)
SET_TIMEOUT_STATE	BOOL	Flag zum manuellen Überspringen der Initialisierung eines CANopen-Slaves, wenn Folgendes zutrifft: • Slave ist nicht im Netzwerk vorhanden • und Slave ist nicht als optional konfiguriert
SET_NODE_STATE	BOOL	Flag zum manuellen Einleiten der Initialisierung eines CANopen-Slaves Der Slave hatte sich beim Zugriff auf das Objekt 0x1000 als ein anderer Gerätetyp identifiert, als in der EDS-Datei angegeben ist, die in der CODESYS-Steuerungskonfiguration eingebunden wurde

Beispiel: CANx_MASTER_STATUS

2031

Slave-Informationen

2699

Damit Sie auf die Informationen der einzelnen CANopen-Knoten zugreifen können, müssen Sie ein Array der jeweiligen Struktur anlegen. Die Strukturen sind in der Bibliothek enthalten. Sie können Sie im Bibliotheksverwalter unter [Datentypen] sehen.

Die Anzahl der Array-Elemente wird bestimmt durch die Globale Variable MAX_NODEINDEX, die automatisch vom CANopen-Stack angelegt wird. Sie enthält die Anzahl der im Netzwerkkonfigurator angegebenen Slaves minus 1.

> ! Die Nummern der Array-Elemente entsprechen **nicht** der Node-ID. Der Identifier kann aus der jeweiligen Struktur unter NODE_ID ausgelesen werden.

Programm-Beispiel zu CAN1_MASTER_STATUS

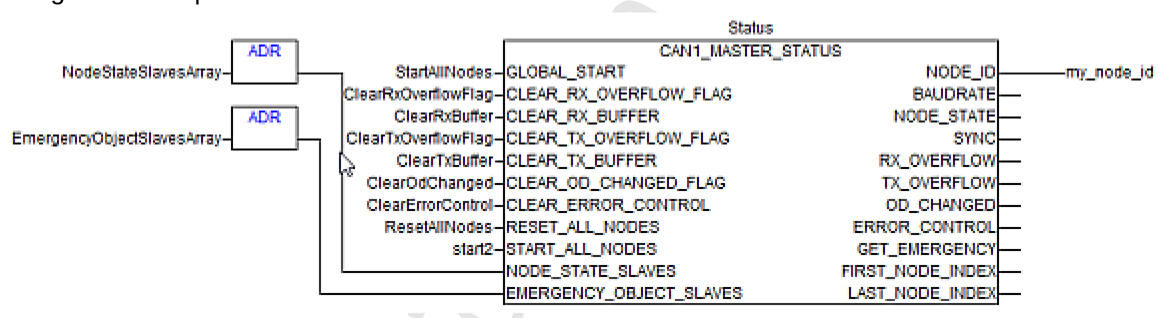
20651

Variablen-Deklaration:

```
VAR
    Status: CAN1_MASTER_STATUS;


    LedStatus: BOOL:= TRUE;
    StartAllNodes: BOOL:= TRUE;
    ClearRxOverflowFlag: BOOL;
    ClearRxBuffer: BOOL;
    ClearTxOverflowFlag: BOOL;
    ClearTxBuffer: BOOL;
    ClearOdChanged: BOOL;
    ClearErrorControl: BOOL;
    ResetAllNodes: BOOL;
    NodeStateSlavesArray: ARRAY [0..MAX_NODEINDEX] OF CAN1_NODE_STATE;
    EmergencyObjectSlavesArray: ARRAY[0..MAX_NODEINDEX] OF CAN1_EMERGENCY_MESSAGE;
    my_node_id: BYTE;
    my_baudrate: WORD;
    my_node_state: INT;
    Sync: BOOL;
    RxOverflow: BOOL;
    TxOverflow: BOOL;
    OdChanged: BOOL;
    GuardHeartbeatErrorArray: ARRAY[0..7] OF BYTE;
    GetEmergency: CAN1_EMERGENCY_MESSAGE;
    start2: BOOL;
    Ency_handler: CAN1_MASTER_EMCY_HANDLER;
    reset_emcy: BOOL;
END_VAR
```

Programm-Beispiel:



Struktur Knoten-Status

2034

```
TYPE CAN1_NODE_STATE :
STRUCT
    NODE_ID: BYTE;
    NODE_STATE: BYTE;
    LAST_STATE: BYTE;
    RESET_NODE: BOOL;
    START_NODE: BOOL;
    PREOP_NODE: BOOL;
    SET_TIMEOUT_STATE: BOOL;
    SET_NODE_STATE: BOOL;
END_STRUCT
END_TYPE
```

Struktur Emergency_Message

2035

```
TYPE CAN1_EMERGENCY_MESSAGE :
STRUCT
    NODE_ID: BYTE;
    ERROR_CODE: WORD;
    ERROR_REGISTER: BYTE;
    MANUFACTURER_ERROR_FIELD: ARRAY[0..4] OF BYTE;
END_STRUCT
END_TYPE
```

Zugriff auf die Strukturen zur Laufzeit der Anwendung

2036

Zur Laufzeit können Sie auf das jeweilige Array-Element über die globalen Variablen der Bibliothek zugreifen und so den Status oder die EMCY-Nachrichten auslesen oder den Knoten zurücksetzen.

```
0001 ⊟--NodeStateList
0002     ⊟--NodeStateList[0]
0003         |---.NODE_ID = 16#02
0004         |---.NODE_STATE = 16#04
0005         |---.LAST_STATE = 16#00
0006         |---.RESET_NODE = FALSE <:= TRUE>
0007         |---.START_NODE = FALSE
0008         |---.PREOP_NODE = FALSE
0009         |---.SET_TIMEOUT_STATE = FALSE
0010         '---.SET_NODE_STATE = FALSE
0011     ⊟--NodeStateList[1]
0012         |---.NODE_ID = 16#03
0013         |---.NODE_STATE = 16#03
0014         |---.LAST_STATE = 16#00
0015         |---.RESET_NODE = FALSE
0016         |---.START_NODE = FALSE
0017         |---.PREOP_NODE = FALSE
0018         |---.SET_TIMEOUT_STATE = FALSE
0019         '---.SET_NODE_STATE = FALSE
0020 ⊟--NodeEmergencyList
0021     ⊟--NodeEmergencyList[0]
0022         |---.NODE_ID = 16#02
0023         |---.ERROR_CODE = 16#0000
0024         |---.ERROR_REGISTER = 16#00
0025         ⊟--.MANUFACTURER_ERROR_FIELD
0026             |---.MANUFACTURER_ERROR_FIELD[0] = 16#00
0027             |---.MANUFACTURER_ERROR_FIELD[1] = 16#00
0028             |---.MANUFACTURER_ERROR_FIELD[2] = 16#00
0029             |---.MANUFACTURER_ERROR_FIELD[3] = 16#00
0030             '---.MANUFACTURER_ERROR_FIELD[4] = 16#00
0031     ⊟--NodeEmergencyList[1]
0032         |---.NODE_ID = 16#03
0033         |---.ERROR_CODE = 16#0000
0034         |---.ERROR_REGISTER = 16#00
0035         ⊟--.MANUFACTURER_ERROR_FIELD
0036             |---.MANUFACTURER_ERROR_FIELD[0] = 16#00
0037             |---.MANUFACTURER_ERROR_FIELD[1] = 16#00
0038             |---.MANUFACTURER_ERROR_FIELD[2] = 16#00
0039             |---.MANUFACTURER_ERROR_FIELD[3] = 16#00
0040             '---.MANUFACTURER_ERROR_FIELD[4] = 16#00
```

Setzen Sie im obigen Beispiel ResetSingleNodeArray[0].RESET_NODE kurzzeitig auf TRUE, wird der erste Knoten im Konfigurationsbaum zurückgesetzt.

ⓘ zu den möglichen Fehler-Codes: → Systemhandbuch "Know-How ecomat*mobile*"
→ Kapitel *CAN / CANopen: Fehler und Fehlerbehandlung*.

5.2.3 Bausteine: CANopen-Slave

Inhalt

1874

Für den CANopen-Slave stellt **ifm electronic** eine Reihe von Bausteinen zur Verfügung, die im Folgenden erklärt werden.

CANx_SLAVE_EMCY_HANDLER

2050

x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_CANopenxSlave_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

2053

CANx_SLAVE_EMCY_HANDLER verwaltet den geräteeigenen Fehlerstatus des CANopen-Slaves:
- Error Register (Index 0x1001) und
- Error Field (Index 0x1003) des CANopen Objektverzeichnis.

► Den FB in folgenden Fällen aufrufen:
- der Fehlerstatus soll ins CAN-Netzwerk übertragen werden und
- die Fehlernachrichten des Anwendungsprogramms sollen im Objektverzeichnis gespeichert werden.

! Sollen die Fehlernachrichten im Objektverzeichnis gespeichert werden?

► **Nach** dem (mehrfachen) Bearbeiten von *CANx_SLAVE_SEND_EMERGENCY* (→ Seite 96) einmalig CANx_SLAVE_EMCY_HANDLER aufrufen!

Parameter der Eingänge

2054

Parameter	Datentyp	Beschreibung
CLEAR_ERROR_FIELD	BOOL	FALSE ⇨ TRUE (Flanke): • Inhalt des ERROR_FIELD an FB-Ausgang ausgeben • Inhalt des ERROR_FIELD im Objektverzeichnis löschen sonst: diese Funktion wird nicht ausgeführt

Parameter der Ausgänge

2055

Parameter	Datentyp	Beschreibung
ERROR_REGISTER	BYTE	Zeigt den Inhalt des OBV Index 0x1001 (Error-Register)
ERROR_FIELD	ARRAY [0..5] OF WORD	Zeigt den Inhalt des OBV Index 0x1003 (Error-Field) ERROR_FIELD[0]: Anzahl der gespeicherten Fehler ERROR_FIELD[1...5]: gespeicherte Fehler, der jüngste Fehler steht im Index [1]

CANx_SLAVE_NODEID

2044

= CANx Slave Node-ID

x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_CANopenxSlave_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

2049

CANx_SLAVE_NODEID ermöglicht das Einstellen der Node-ID eines CANopen-Slaves zur Laufzeit des Anwendungsprogramms.

Der FB wird im Normalfall bei der Initialisierung der Steuerung einmalig, im ersten Zyklus, aufgerufen. Anschließend wird der Eingang ENABLE wieder auf FALSE gesetzt.

Parameter der Eingänge

2047

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	FALSE ⇨ TRUE (Flanke): Parameter übernehmen und aktivieren sonst: diese Funktion wird nicht ausgeführt
NODEID	BYTE	Node-ID = ID des Knotens zulässige Werte = 0...127

CANx_SLAVE_SEND_EMERGENCY

2056

x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_CANopenxSlave_Vxxyyzz.LIB`

Symbol in CODESYS:

```
CANx_SLAVE_SEND_EMERGENCY
ENABLE
ERROR
ERROR_CODE
ERROR_REGISTER
MANUFACTURER_ERROR_FIELD
```

Beschreibung

2059

CANx_SLAVE_SEND_EMERGENCY versendet anwendungsspezifische Fehlerstatus. Das sind Fehlernachrichten, die zusätzlich zu den geräteinternen Fehlernachrichten (z.B. Kurzschluss am Ausgang) gesendet werden sollen.

► Den FB aufrufen, wenn der Fehlerstatus an andere Geräte im Netzwerkverbund übertragen werden soll.

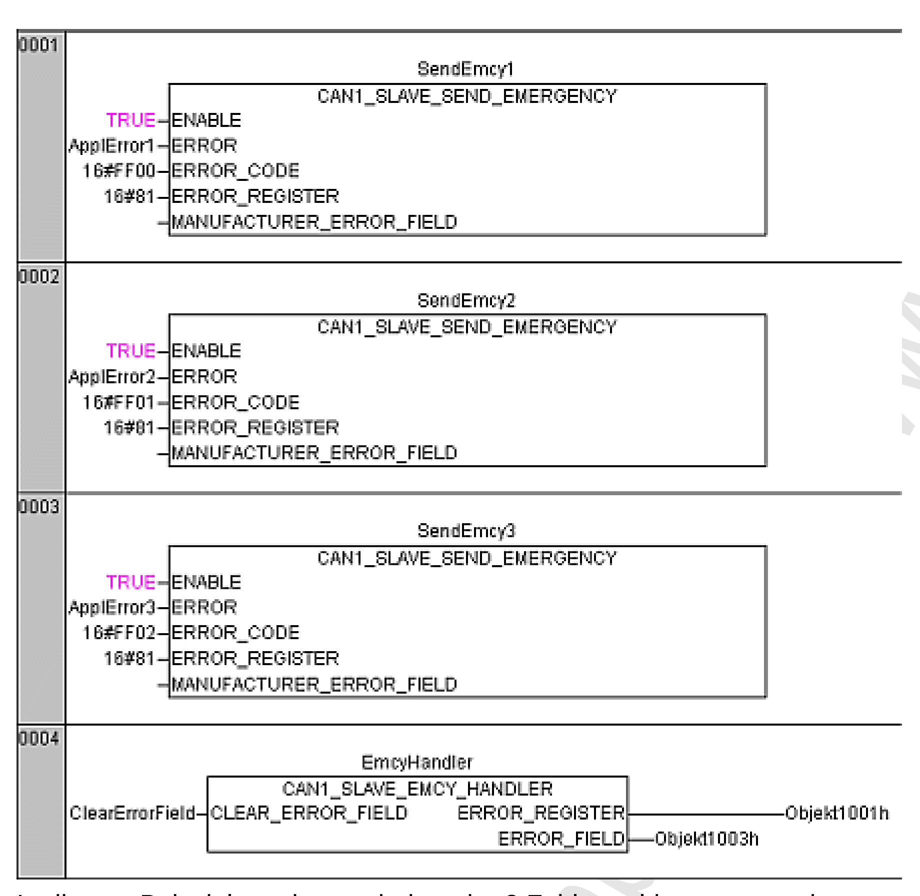
Parameter der Eingänge

2060

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
ERROR	BOOL	Über diesen Eingang wird dem FB die Information übergeben, ob der zum konfigurierten Fehlercode gehörende Fehler aktuell anliegt. FALSE ⇨ TRUE (Flanke): sendet den anstehenden Fehler-Code falls Eingang in der letzten Sekunde nicht TRUE war TRUE ⇨ FALSE (Flanke) UND Fehler steht nicht mehr an: Nach Verzögerung von ca. 1 s: > Null-Fehlermeldung wird gesendet sonst: diese Funktion wird nicht ausgeführt
ERROR_CODE	WORD	Der Error-Code gibt detailliert Auskunft über den erkannten Fehler. Die Werte sollten gemäß der CANopen-Spezifikation eingetragen werden.
ERROR_REGISTER	BYTE	ERROR_REGISTER gibt die Art des Fehlers an. Der hier angegebene Wert wird mit allen anderen aktuell aktiven Fehlernachrichten bitweise ODER-verknüpft. Der sich hierbei ergebende Wert wird ins Error-Register (Index 1001_{16}/00) geschrieben und mit der EMCY-Nachricht versendet. Die Werte sollten gemäß der CANopen-Spezifikation eingetragen werden.
MANUFACTURER_ERROR_FIELD	ARRAY [0..4] OF BYTE	Hier können bis zu 5 Bytes anwendungsspezifische Fehlerinformationen eingetragen werden. Das Format ist dabei frei wählbar.

Beispiel: CANx_SLAVE_SEND_EMERGENCY

2062



In diesem Beispiel werden nacheinander 3 Fehlermeldungen generiert:

1. ApplError1, Code = 0xFF00 im Fehlerregister 0x81
2. ApplError2, Code = 0xFF01 im Fehlerregister 0x81
3. ApplError3, Code = 0xFF02 im Fehlerregister 0x81

Der FB CAN1_SLAVE_EMCY_HANDLER sendet die Fehlermeldungen an das Fehler-Register "Objekt 0x1001" im Fehler-Array "Objekt 0x1003".

CANx_SLAVE_SET_PREOP

2700

x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_CANopenxSlave_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

2703

CANx_SLAVE_SET_PREOP schaltet den Betriebsmodus dieses CANopen-Slaves von OPERATIONAL auf PRE-OPERATIONAL.

Normalerweise schaltet das Gerät im Fehlerfall lediglich die Ausgänge ab. Unter bestimmten Bedingungen kann es erforderlich sein, dass das Anwendungsprogramm den Betriebszustand des als Slave arbeitenden Geräts auf PRE-OPERATIONAL setzt. Dies erfolgt über den hier beschriebenen FB.

Parameter der Eingänge

2704

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	FALSE ⇨ TRUE (Flanke): Slave auf PRE-OPERATIONAL setzen sonst: diese Funktion wird nicht ausgeführt

CANx_SLAVE_STATUS

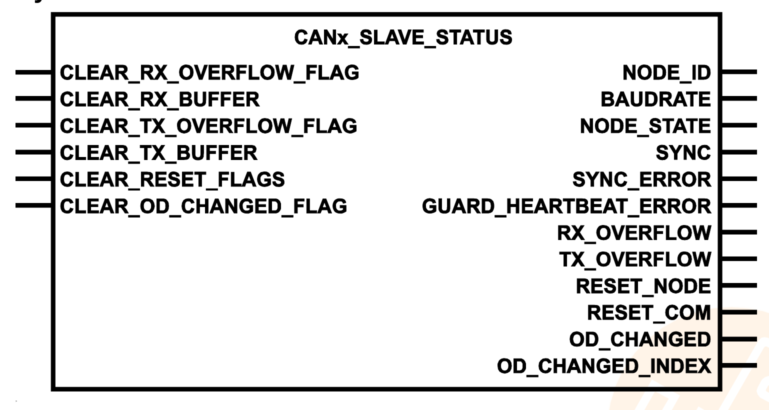
2706

x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_CANopenxSlave_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

2707

CANx_SLAVE_STATUS zeigt den Status des als CANopen-Slave eingesetzten Gerätes.

(!) Wir empfehlen dringend, die Auswertung des Netzwerkstatus über diesen FB vorzunehmen.

Parameter der Eingänge

2708

Parameter	Datentyp	Beschreibung
CLEAR_RX_OVERFLOW_FLAG	BOOL	FALSE ⇨ TRUE (Flanke): Fehlerflag RX_OVERFLOW löschen sonst: diese Funktion wird nicht ausgeführt
CLEAR_RX_BUFFER	BOOL	FALSE ⇨ TRUE (Flanke): Daten im Empfangspuffer löschen sonst: diese Funktion wird nicht ausgeführt
CLEAR_TX_OVERFLOW_FLAG	BOOL	FALSE ⇨ TRUE (Flanke): Fehlerflag TX_OVERFLOW löschen sonst: diese Funktion wird nicht ausgeführt
CLEAR_TX_BUFFER	BOOL	FALSE ⇨ TRUE (Flanke): Daten im Sendepuffer löschen sonst: diese Funktion wird nicht ausgeführt
CLEAR_RESET_FLAGS	BOOL	FALSE ⇨ TRUE (Flanke): Flag RESET_NODE löschen Flag RESET_COM löschen sonst: diese Funktion wird nicht ausgeführt
CLEAR_OD_CHANGED_FLAGS	BOOL	FALSE ⇨ TRUE (Flanke): Flag OD_CHANGED löschen Flag OD_CHANGED_INDEX löschen sonst: diese Funktion wird nicht ausgeführt

Parameter der Ausgänge

2068

Parameter	Datentyp	Beschreibung
NODE_ID	BYTE	aktuelle Knoten-ID des CANopen-Slaves
BAUDRATE	WORD	aktuelle Baudrate des CANopen-Knotens in [kBaud]
NODE_STATE	BYTE	aktueller Status des CANopen-Slaves 0 = Bootup-Nachricht versendet 4 = CANopen-Slave im Status PRE-OPERATIONAL und wird per SDO-Zugriff konfiguriert 5 = CANopen-Slave im Status OPERATIONAL 127 = CANopen-Slave im Status PRE-OPERATIONAL
SYNC	BOOL	SYNC-Signal des CANopen-Masters TRUE: Im letzten Zyklus wurde ein SYNC-Signal empfangen FALSE: Im letzten Zyklus wurde kein SYNC-Signal empfangen
SYNC_ERROR	BOOL	TRUE: Fehler: das SYNC-Signal des Masters wurde nicht oder zu spät (nach Ablauf von ComCyclePeriod) empfangen FALSE: kein SYNC-Fehler
GUARD_HEARTBEAT_ERROR	BOOL	TRUE: Fehler: das Guarding- oder Heartbeat-Signal des Masters wurde nicht oder zu spät empfangen FALSE: kein Guarding- oder Heartbeat-Fehler
RX_OVERFLOW	BOOL	TRUE: Fehler: Empfangspuffer-Überlauf FALSE: kein Überlauf
TX_OVERFLOW	BOOL	TRUE: Fehler: Sendepuffer-Überlauf FALSE: kein Überlauf
RESET_NODE	BOOL	TRUE: CANopen-Stack des Slaves vom Master zurückgesetzt FALSE: CANopen-Stack des Slaves nicht zurückgesetzt
RESET_COM	BOOL	TRUE: Kommunikations-Interface des CAN-Stack wurde vom Master zurückgesetzt FALSE: Kommunikations-Interface nicht zurückgesetzt
OD_CHANGED	BOOL	TRUE: Daten im Objektverzeichnis des CANopen-Masters wurden geändert FALSE: keine Datenänderung
OD_CHANGED_INDEX	INT	Index des zuletzt geänderten Objektverzeichnis-Eintrags

5.2.4 Bausteine: CANopen SDOs

Inhalt

2071

Hier finden Sie **ifm**-Bausteine für den Umgang von CANopen mit Service Data Objects (SDOs).

CANx_SDO_READ

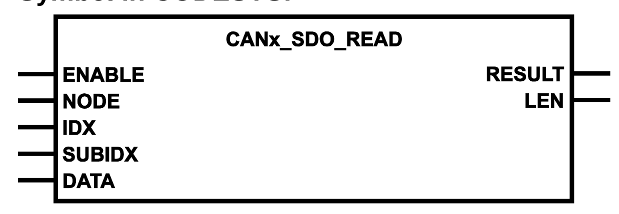
621

x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

624

CANx_SDO_READ liest das →*SDO* (→ Seite 325) mit den angegebenen Indizes aus dem Knoten aus.

Voraussetzung: Knoten muss sich im Zustand PRE-OPERATIONAL oder OPERATIONAL befinden.

Über diese Indizes können die Einträge im Objektverzeichnis gelesen werden. Dadurch ist es möglich, die Knotenparameter gezielt zu lesen.

Beispiel:

0004
sdo1_data
ADR
SDO_read1
CAN1_SDO_READ
m1 ENABLE RESULT
node NODE LEN LEN1
idx IDX
subidx SUBIDX
DATA
EQ
R
m1

Parameter der Eingänge

625

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
NODE	BYTE	CANopen-ID des Knotens zulässig = 1...127 = 0x01...0x7F
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex bezogen auf den Index im Objektverzeichnis
DATA	DWORD	Adresse des Empfangsdaten-Arrays zulässige Länge = 0...255 ! Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!

Parameter der Ausgänge

626

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)
LEN	WORD	Länge des Eintrags in "Anzahl der Bytes" Der Wert für LEN darf nicht größer sein als die Größe des Empfangs-Arrays. Andernfalls werden beliebige Daten in der Anwendung überschrieben.

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
2	02	Funktionsbaustein ist aktiv (Aktion noch nicht beendet)
3	03	Fehler, keine Daten während der Überwachungszeit empfangen

CANx_SDO_WRITE

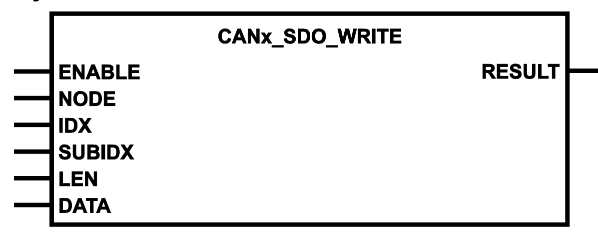
615

x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

618

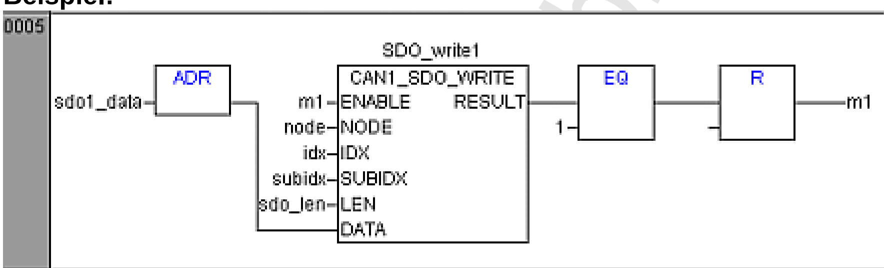
CANx_SDO_WRITE schreibt das →*SDO* (→ Seite 325) mit den angegebenen Indizes in den Knoten.

Voraussetzung: Knoten muss sich im Zustand PRE-OPERATIONAL oder OPERATIONAL befinden.

Über diesen FB können die Einträge im Objektverzeichnis geschrieben werden. Dadurch ist es möglich, die Knotenparameter gezielt zu setzen.

> ! Der Wert für LEN muss kleiner sein als die Größe des Sende-Arrays. Andernfalls werden beliebige Daten versendet.

Beispiel:



Parameter der Eingänge

619

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
NODE	BYTE	CANopen-ID des Knotens zulässig = 1...127 = 0x01...0x7F
IDX	WORD	Index im Objektverzeichnis
SUBIDX	BYTE	Subindex bezogen auf den Index im Objektverzeichnis
LEN	WORD	Länge des Eintrags in "Anzahl der Bytes" Der Wert für LEN darf nicht größer sein als die Größe des Sende-Arrays. Andernfalls werden beliebige Daten versendet.
DATA	DWORD	Adresse des Sendedaten-Arrays zulässige Länge = 0...255 [!] Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!

Parameter der Ausgänge

620

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
2	02	Funktionsbaustein ist aktiv (Aktion noch nicht beendet)
3	03	Fehler, Daten können nicht übertragen werden

5.2.5 Bausteine: SAE J1939

Inhalt

2273

Für SAE J1939 stellt **ifm electronic** eine Reihe von Bausteinen zur Verfügung, die im Folgenden erklärt werden.

J1939_x

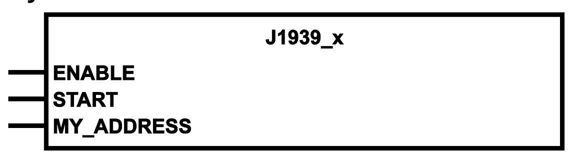
2274

x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_J1939_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

2276

J1939_x dient als Protokoll-Handler für das Kommunikationsprofil SAE J1939.

Zur Abwicklung der Kommunikation muss der Protokoll-Handler in jedem Programmzyklus aufgerufen werden. Dazu wird der Eingang ENABLE auf TRUE gesetzt.

! Einmal gesetzt, muss ENABLE auf TRUE bleiben!

Der Protokoll-Handler wird gestartet, wenn der Eingang START für einen Zyklus auf TRUE gesetzt wird.

Über MY_ADRESS wird dem Controller eine Geräteadresse übergeben. Sie muss sich von Adressen der anderen J1939-Busteilnehmer unterscheiden. Sie kann dann von anderen Busteilnehmern ausgelesen werden.

Parameter der Eingänge

469

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
START	BOOL	TRUE (nur 1 Zyklus lang): J1939-Protokoll an CAN-Schnittstelle x starten FALSE: im weiteren Programmablauf
MY_ADDRESS	BYTE	J1939-Adresse des Geräts

J1939_x_GLOBAL_REQUEST

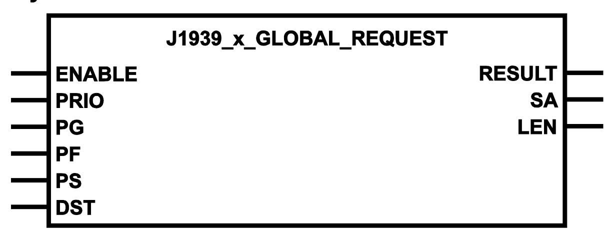
2282

x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_J1939_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

2301

J1939_x_GLOBAL_REQUEST ist für das automatische Anfordern einzelner Nachrichten von allen (global) aktiven J1939-Netzwerkteilnehmern verantwortlich. Dazu werden dem FB die Parameter PG, PF, PS und die Adresse des Arrays DST übergeben, in dem die empfangenen Daten abgelegt werden.

Info

PGN = [Page] + [PF] + [PS]
PDU = [PRIO] + [PGN] + [J1939-Adresse] + [Daten]

13790

ACHTUNG

Daten können unzulässig überschrieben werden!

► Ein Empfangs-Array mit einer Größe von 1 785 Bytes anlegen!
Dies ist die maximale Größe einer J1939-Nachricht.

► Die Anzahl empfangener Daten prüfen:
der Wert darf nicht größer sein als das bereitgestellte Empfangs-Array!

► Für jede angefragte Nachricht eine eigene Instanz des FBs verwenden!

► Für die Zieladresse DST gilt:
[!] Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!

► Zusätzlich die Priorität (typisch 3, 6 oder 7) übergeben.

► Da das Anfordern der Daten über mehrere Steuerungszyklen abgewickelt werden kann, muss dieser Vorgang über das RESULT-Byte ausgewertet werden.

- RESULT = 2: der Baustein wartet auf Daten der Teilnehmer.
- RESULT = 1: von einem Teilnehmer wurden Daten empfangen.
Der Ausgang LEN zeigt an, wie viele Datenbytes empfangen wurden.
Diese neuen Daten in DST sofort speichern / auswerten!
Der Empfang einer weiteren Nachricht überschreibt die Daten auf der Speicheradresse DST.
- RESULT = 0: innerhalb von 1,25 Sekunden hat kein Teilnehmer am Bus eine Antwort gesendet.
Der Baustein wird wieder inaktiv.
Erst jetzt darf ENABLE wieder auf FALSE gesetzt werden!

► Für das Empfangen von Daten von mehreren Teilnehmern in schneller Folge:
den Baustein im selben SPS-Zyklus mehrmals aufrufen und direkt auswerten!

Parameter der Eingänge

463

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
PRIO	BYTE	Nachrichten-Prioritätin der PDU (Parameter Data Unit) zulässig = 0…7
PG	BYTE	Data Page Wert der definierten PGN (Parameter Group Number) zulässig = 0...1 (normalerweise = 0)
PF	BYTE	PDU format byte Wert der definierten PGN (Parameter Group Number) PDU2 (global) = 240...255
PS	BYTE	PDU specific byte Wert der definierten PGN (Parameter Group Number) GE (Group Extension) = 0...255
DST	DWORD	Startadresse im Zielspeicher ! Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!

Info

PGN = [Page] + [PF] + [PS]
PDU = [PRIO] + [PGN] + [J1939-Adresse] + [Daten]

Parameter der Ausgänge

464

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)
SA	BYTE	J1939-Adresse des antwortenden Geräts
LEN	WORD	Anzahl der empfangenen Bytes

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
2	02	Funktionsbaustein ist aktiv (Aktion noch nicht beendet)
3	03	Fehler

J1939_x_RECEIVE

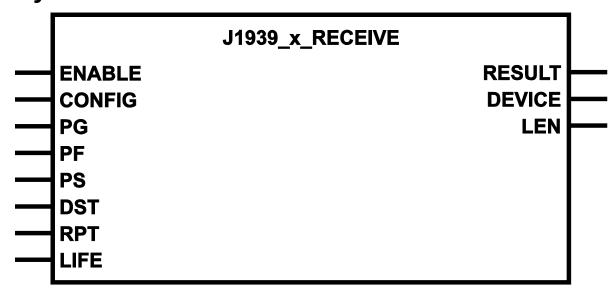
2278

x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_J1939_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

2288

J1939_x_RECEIVE dient dem Empfang einer einzelnen Nachricht oder eines Nachrichtenblocks.

Dazu muss der FB über den Eingang CONFIG für einen Zyklus initialisiert werden. Bei der Initialisierung werden die Parameter PG, PF, PS, RPT, LIFE und die Speicheradresse des Datenarrays DST übergeben.

> [!] Nach dem ersten Konfigurieren können diese Parameter im laufenden Anwendungsprogramm nicht mehr verändert werden: PG, PF, PS, RPT, LIFE, DST.

13790

ACHTUNG

Daten können unzulässig überschrieben werden!

- ► Ein Empfangs-Array mit einer Größe von 1 785 Bytes anlegen!
  Dies ist die maximale Größe einer J1939-Nachricht.
- ► Die Anzahl empfangener Daten prüfen:
  der Wert darf nicht größer sein als das bereitgestellte Empfangs-Array!

- ► Für die Zieladresse DST gilt:
  [!] Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!

[!] Nach dem ersten Setzen kann RPT nicht mehr verändert werden!

- ► Der Datenempfang muss über das RESULT-Byte ausgewertet werden. Wird RESULT = 1, können die Daten von der über DST übergebenen Speicheradresse ausgelesen und weiter verarbeitet werden.
- > Der Empfang einer neuen Nachricht überschreibt die Daten auf der Speicheradresse DST.
- > Die Anzahl der empfangenen Nachrichten-Bytes wird über den Ausgang LEN angezeigt.
- > Wird RESULT = 3, wurden im angegebenen Zeitfenster (LIFE • RPT) keine gültigen Nachrichten empfangen.

> [!] Dieser Baustein muss auch eingesetzt werden, wenn die Nachrichten mit den FBs J1939_..._REQUEST angefordert werden.

Parameter der Eingänge

457

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CONFIG	BOOL	TRUE (im 1. Zyklus): Datenobjekt konfigurieren FALSE: im weiteren Programmablauf
PG	BYTE	Data Page Wert der definierten PGN (Parameter Group Number) zulässig = 0...1 (normalerweise = 0)
PF	BYTE	PDU format byte Wert der definierten PGN (Parameter Group Number) PDU1 (specific) = 0...239 PDU2 (global) = 240...255
PS	BYTE	PDU specific byte Wert der definierten PGN (Parameter Group Number) Wenn PF = PDU1 ⇨ PS = DA (Destination Address) (DA = J1939-Adresse des externen Geräts) Wenn PF = PDU2 ⇨ PS = GE (Group Extension)
DST	DWORD	Startadresse im Zielspeicher [!] Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!
RPT	TIME	Überwachungszeit Innerhalb dieses angegebenen Zeitfensters müssen die Telegramme zyklisch empfangen werden. > Andernfalls erfolgt eine Fehlermeldung. RPT = T#0s ⇨ keine Überwachung [!] Nach dem ersten Setzen kann RPT nicht mehr verändert werden!
LIFE	BYTE	tolerierte Anzahl der nicht empfangenen J1939-Nachrichten

Parameter der Ausgänge

458

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)
DEVICE	BYTE	J1939-Adresse des Absenders
LEN	WORD	Anzahl der empfangenen Bytes

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
3	03	Fehler, keine Daten während der Überwachungszeit empfangen

J1939_x_RESPONSE

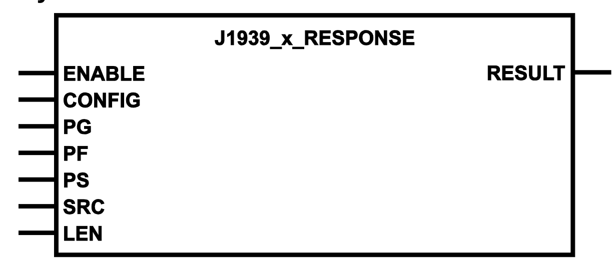
2280

x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_J1939_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

2299

J1939_x_RESPONSE organisiert die automatische Antwort auf ein Request-Telegramm (Anforderungstelegramm).

Der FB ist für das automatische Versenden von Nachrichten auf "Global Requests" und "Specific Requests" verantwortlich. Dazu muss der FB über den Eingang CONFIG für einen Zyklus initialisiert werden.

Dem FB werden die Parameter PG, PF, PS, RPT und die Adresse des Datenarrays SRC übergeben.

► Für die Quelladresse SRC gilt:
  (!) Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!

► Zusätzlich die Anzahl der zu übertragenen Datenbytes übergeben.

Parameter der Eingänge

451

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
CONFIG	BOOL	TRUE (im 1. Zyklus): Datenobjekt konfigurieren FALSE: im weiteren Programmablauf
PG	BYTE	Data Page Wert der definierten PGN (Parameter Group Number) zulässig = 0...1 (normalerweise = 0)
PF	BYTE	PDU format byte Wert der definierten PGN (Parameter Group Number) PDU1 (specific) = 0...239 PDU2 (global) = 240...255
PS	BYTE	PDU specific byte Wert der definierten PGN (Parameter Group Number) Wenn PF = PDU1 ⇨ PS = DA (Destination Address) (DA = J1939-Adresse des externen Geräts) Wenn PF = PDU2 ⇨ PS = GE (Group Extension)
SRC	DWORD	Startadresse im Quellspeicher (!) Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!
LEN	WORD	Anzahl (≥ 1) der zu übertragenden Daten-Bytes

Parameter der Ausgänge

13993

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	Datenübertragung wurde ohne Fehler beendet
2	02	Funktionsbaustein ist aktiv (Aktion noch nicht beendet)
3	03	Fehler, Daten können nicht übertragen werden

J1939_x_SPECIFIC_REQUEST

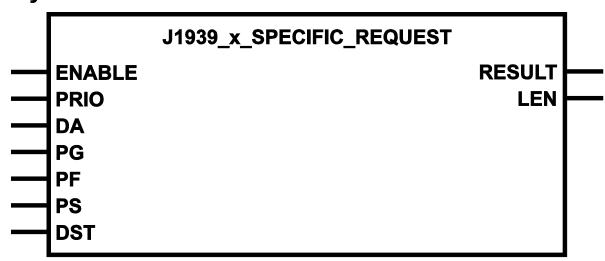
2281

x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_J1939_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

2300

J1939_x_SPECIFIC_REQUEST ist für das automatische Anfordern einzelner Nachrichten von einem bestimmten (specific) J1939-Netzwerkteilnehmer verantwortlich. Dazu werden dem FB die logische Geräteadresse DA, die Parameter PG, PF, PS und die Adresse des Arrays DST übergeben, in dem die empfangenen Daten abgelegt werden.

Info

PGN = [Page] + [PF] + [PS]
PDU = [PRIO] + [PGN] + [J1939-Adresse] + [Daten]

13790

ACHTUNG

Daten können unzulässig überschrieben werden!

► Ein Empfangs-Array mit einer Größe von 1 785 Bytes anlegen!
Dies ist die maximale Größe einer J1939-Nachricht.

► Die Anzahl empfangener Daten prüfen:
der Wert darf nicht größer sein als das bereitgestellte Empfangs-Array!

► Für die Zieladresse gilt:
(!) Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!

► Zusätzlich die Priorität (typisch 3, 6 oder 7) übergeben.

► Da das Anfordern der Daten über mehrere Steuerungszyklen abgewickelt werden kann, muss dieser Vorgang über das RESULT-Byte ausgewertet werden. Wird RESULT = 1, wurden alle Daten empfangen.

> Der Ausgang LEN zeigt an, wie viele Datenbytes empfangen wurden.

> Wird innerhalb von 1,25 Sekunden vom angeforderten Teilnehmer keine Antwort gesendet, meldet der FB einen Fehler (⇨ RESULT = 3).

Parameter der Eingänge

445

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
PRIO	BYTE	Nachrichten-Prioritätin der PDU (Parameter Data Unit) zulässig = 0…7
DA	BYTE	J1939-Adresse des angefragten Geräts
PG	BYTE	Data Page Wert der definierten PGN (Parameter Group Number) zulässig = 0...1 (normalerweise = 0)
PF	BYTE	PDU format byte Wert der definierten PGN (Parameter Group Number) PDU1 (specific) = 0...239 PDU2 (global) = 240...255
PS	BYTE	PDU specific byte Wert der definierten PGN (Parameter Group Number) Wenn PF = PDU1 ⇨ PS = DA (Destination Address) (DA = J1939-Adresse des externen Geräts) Wenn PF = PDU2 ⇨ PS = GE (Group Extension)
DST	DWORD	Startadresse im Zielspeicher ❗ Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!

Info

PGN = [Page] + [PF] + [PS]
PDU = [PRIO] + [PGN] + [J1939-Adresse] + [Daten]

Parameter der Ausgänge

446

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)
LEN	WORD	Anzahl der empfangenen Bytes

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
2	02	Funktionsbaustein ist aktiv (Aktion noch nicht beendet)
3	03	Fehler

J1939_x_TRANSMIT

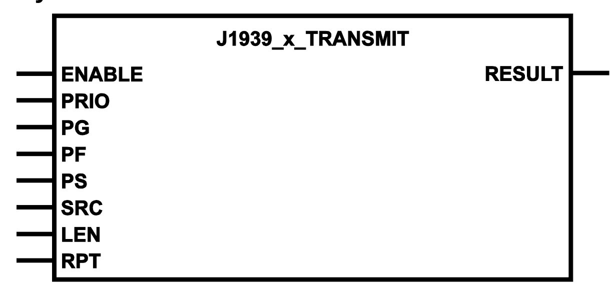
279

x = 1...n = Nummer der CAN-Schnittstelle (je nach Gerät, → Datenblatt)

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_J1939_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

2298

J1939_x_TRANSMIT ist für das Versenden einzelner Nachrichten oder Nachrichtenblocks verantwortlich. Dazu werden dem FB die Parameter PG, PF, PS, RPT und die Adresse des Datenarrays SRC übergeben.

Info

PGN = [Page] + [PF] + [PS]
PDU = [PRIO] + [PGN] + [J1939-Adresse] + [Daten]

► Für die Quelladresse SRC gilt:
  Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!

► Zusätzlich die Anzahl der zu übertragenen Datenbytes und die Priorität (typisch 3, 6 oder 7) übergeben.

► Da das Versenden der Daten über mehrere Steuerungszyklen abgewickelt wird, muss der Vorgang über das RESULT-Byte ausgewertet werden. Wird RESULT = 1, wurden alle Daten übertragen.

Wenn mehr als 8 Bytes gesendet werden sollen, wird ein "multi package transfer" durchgeführt.

Parameter der Eingänge

439

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
PRIO	BYTE	Nachrichten-Prioritätin der PDU (Parameter Data Unit) zulässig = 0…7
PG	BYTE	Data Page Wert der definierten PGN (Parameter Group Number) zulässig = 0...1 (normalerweise = 0)
PF	BYTE	PDU format byte Wert der definierten PGN (Parameter Group Number) PDU1 (specific) = 0...239 PDU2 (global) = 240...255
PS	BYTE	PDU specific byte Wert der definierten PGN (Parameter Group Number) Wenn PF = PDU1 ⇨ PS = DA (Destination Address) (DA = J1939-Adresse des externen Geräts) Wenn PF = PDU2 ⇨ PS = GE (Group Extension)
SRC	DWORD	Startadresse im Quellspeicher Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!
LEN	WORD	Anzahl der zu übertragenden Daten-Bytes zulässig = 1...1 785 = 0x0001...0x06F9
RPT	TIME	Wiederholzeit, innerhalb der die Daten-Telegramme zyklisch versendet werden sollen RPT = T#0s ⇨ nur einmalig versenden

Info

PGN = [Page] + [PF] + [PS]
PDU = [PRIO] + [PGN] + [J1939-Adresse] + [Daten]

Parameter der Ausgänge

440

Parameter	Datentyp	Beschreibung
RESULT	BYTE	Rückmeldung des Funktionsbausteins (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für RESULT:

Wert dez	Wert hex	Beschreibung
0	00	FB ist inaktiv
1	01	FB-Ausführung wurde ohne Fehler beendet – Daten sind gültig
2	02	Funktionsbaustein ist aktiv (Aktion noch nicht beendet)
3	03	Fehler, Daten können nicht übertragen werden

5.2.6 Bausteine: serielle Schnittstelle

Inhalt

13011

HINWEIS

Voreingestellt steht die serielle Schnittstelle dem Anwender nicht zur Verfügung, da sie für den Programm-Download und das Debugging genutzt wird.

Setzt der Anwender das Systemmerkerbit SERIAL_MODE=TRUE, dann kann die Schnittstelle frei genutzt werden. Ein Debugging des Anwendungsprogramms ist dann nur noch über eine der 4 CAN-Schnittstellen oder über USB möglich.

Mit den folgend aufgeführten Bausteinen kann die serielle Schnittstelle im Anwendungsprogramm genutzt werden.

SERIAL_PENDING

314

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

12994

SERIAL_PENDING ermittelt die Anzahl der im seriellen Empfangspuffer gespeicherten Datenbytes.

Im Gegensatz zu *SERIAL_RX* (→ Seite 120) bleibt der Inhalt des Puffers nach Aufruf dieser Funktion unverändert.

Die SERIAL-Bausteine bilden die Grundlage für die Erstellung eines anwendungsspezifischen Protokolls für die serielle Schnittstelle.
Dazu das Systemmerkerbit SERIAL_MODE=TRUE setzen!

HINWEIS

Voreingestellt steht die serielle Schnittstelle dem Anwender nicht zur Verfügung, da sie für den Programm-Download und das Debugging genutzt wird.

Setzt der Anwender das Systemmerkerbit SERIAL_MODE=TRUE, dann kann die Schnittstelle frei genutzt werden. Ein Debugging des Anwendungsprogramms ist dann nur noch über eine der 4 CAN-Schnittstellen oder über USB möglich.

Parameter der Ausgänge

12996

Parameter	Datentyp	Beschreibung
NUMBER	WORD	Anzahl der empfangenen Datenbytes (1...1 000)

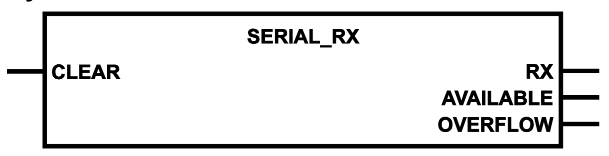
SERIAL_RX

308

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

12997

SERIAL_RX liest mit jedem Aufruf ein empfangenes Datenbyte aus dem seriellen Empfangspuffer aus.

Gehen mehr als 1 000 Datenbytes ein, läuft der Puffer über und es gehen Daten verloren. Dieses wird durch das Bit OVERFLOW angezeigt.

Wird eine 7-Bit-Datenübertragung genutzt, enthält das 8. Bit die Parität und muss gegebenenfalls vom Anwender ausgeblendet werden.

Die SERIAL-Bausteine bilden die Grundlage für die Erstellung eines anwendungsspezifischen Protokolls für die serielle Schnittstelle.
Dazu das Systemmerkerbit SERIAL_MODE=TRUE setzen!

> **HINWEIS**
>
> Voreingestellt steht die serielle Schnittstelle dem Anwender nicht zur Verfügung, da sie für den Programm-Download und das Debugging genutzt wird.
>
> Setzt der Anwender das Systemmerkerbit SERIAL_MODE=TRUE, dann kann die Schnittstelle frei genutzt werden. Ein Debugging des Anwendungsprogramms ist dann nur noch über eine der 4 CAN-Schnittstellen oder über USB möglich.

Parameter der Eingänge

312

Parameter	Datentyp	Beschreibung
CLEAR	BOOL	TRUE: Empfangspuffer löschen FALSE: Funktion wird nicht ausgeführt

Parameter der Ausgänge

12931

Parameter	Datentyp	Beschreibung
RX	BYTE	empfangene Byte-Daten aus dem Empfangspuffer
AVAILABLE	WORD	Anzahl der empfangenen Bytes, die sich im Empfangspuffer befinden VOR dem Aufruf des FBs: 0 = keine Daten empfangen 1...1 000 = Anzahl von Bytes im Empfangspuffer
OVERFLOW	BOOL	TRUE: Überlauf des Datenpuffers ⇨ Datenverlust! FALSE: Datenpuffer ist ohne Datenverlust

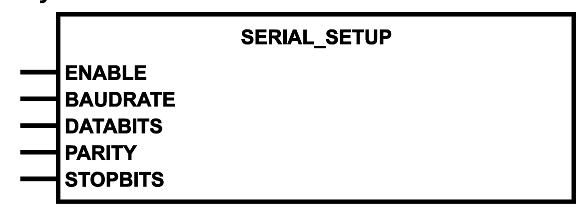
SERIAL_SETUP

302

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

13000

SERIAL_SETUP initialisiert die serielle RS232-Schnittstelle.

Der FB muss nicht zwingend ausgeführt werden, um die serielle Schnittstelle verwenden zu können. Ohne FB-Aufruf gelten die folgend angegebenen Voreinstellungen.

Mit ENABLE=TRUE für einen Zyklus setzt der FB die serielle Schnittstelle auf die angegebenen Parameter. Die mit dem FB vorgenommenen Änderungen werden remanent gespeichert.

> **HINWEIS**
>
> Voreingestellt steht die serielle Schnittstelle dem Anwender nicht zur Verfügung, da sie für den Programm-Download und das Debugging genutzt wird.
>
> Setzt der Anwender das Systemmerkerbit SERIAL_MODE=TRUE, dann kann die Schnittstelle frei genutzt werden. Ein Debugging des Anwendungsprogramms ist dann nur noch über eine der 4 CAN-Schnittstellen oder über USB möglich.

Parameter der Eingänge

13002

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE (nur 1 Zyklus lang): Schnittstelle initialisieren FALSE: im weiteren Programmablauf
BAUDRATE	DWORD	Baudrate zulässige Werte → Datenblatt Voreinstellwert → Datenblatt
DATABITS	BYTE := 8	Anzahl der Daten-Bits zulässig = 7 oder 8
PARITY	BYTE := 0	Parität zulässig: 0=keine, 1=gerade, 2=ungerade Falls DATABITS = 7 und PARITY = 0 parametriert: dann arbeitet der FB mit PARITY = 1
STOPBITS	BYTE := 1	Anzahl der Stopp-Bits zulässig = 1 oder 2

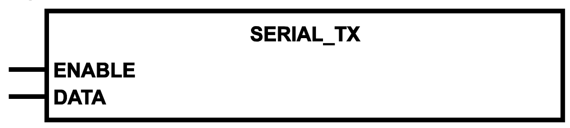
SERIAL_TX

296

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

13003

SERIAL_TX überträgt ein Datenbyte über die serielle RS232-Schnittstelle.

Der FiFo-Sendespeicher fasst 1 000 Bytes.

Mit dem Eingang ENABLE kann die Übertragung freigegeben oder gesperrt werden.

Die SERIAL-Bausteine bilden die Grundlage für die Erstellung eines anwendungsspezifischen Protokolls für die serielle Schnittstelle.
Dazu das Systemmerkerbit SERIAL_MODE=TRUE setzen!

HINWEIS

Voreingestellt steht die serielle Schnittstelle dem Anwender nicht zur Verfügung, da sie für den Programm-Download und das Debugging genutzt wird.

Setzt der Anwender das Systemmerkerbit SERIAL_MODE=TRUE, dann kann die Schnittstelle frei genutzt werden. Ein Debugging des Anwendungsprogramms ist dann nur noch über eine der 4 CAN-Schnittstellen oder über USB möglich.

Parameter der Eingänge

300

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
DATA	BYTE	zu übertragender Wert

5.2.7 Bausteine: SPS-Zyklus optimieren

Inhalt

8609

Hier zeigen wir Ihnen Funktionen zum Optimieren des SPS-Zyklus.

Bausteine: Interrupts verarbeiten

Inhalt

1599

Die SPS arbeitet das gespeicherte Anwendungsprogramm zyklisch in voller Länge ab. Von z.B. äußeren Ereignissen abhängige Verzweigungen im Programm (= bedingte Sprünge) lassen die Zykluszeit variieren. Für bestimmte Funktionen kann dieses Verhalten nachteilig sein.

Mit Hilfe gezielter Unterbrechungen (= Interrupts) des zyklischen Programmablaufs können zeitkritische Abläufe unabhängig vom Zyklus in festen Zeitrastern oder bei bestimmten Ereignissen aufgerufen werden.

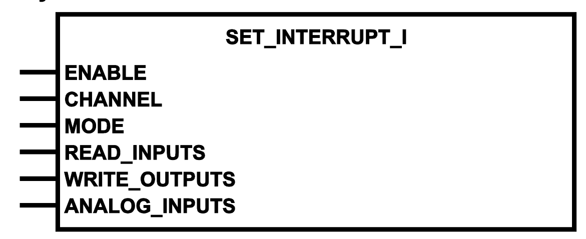
SET_INTERRUPT_I

2381

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

19361
11573

SET_INTERRUPT_I organisiert das Ausführen eines Programmteils durch eine Interrupt-Anforderung über einen Eingangskanal.

In der klassischen SPS ist die Zykluszeit das Maß der Dinge für Echtzeitbetrachtungen. Gegenüber kundenspezifischen Steuerungen ist die SPS damit im Nachteil. Auch ein "Echtzeit-Betriebssystem" ändert nichts an dieser Tatsache, wenn das gesamte Anwendungsprogramm in einem einzigen unveränderlichen Block abläuft.

Ein möglicher Lösungsansatz wäre, die Zykluszeit kurz zu halten. Dieser Weg führt oft dazu, die Anwendung auf mehrere Steuerungszyklen zu verteilen. Die Programmierung wird dadurch jedoch unübersichtlich und schwierig.

Eine andere Möglichkeit besteht darin, einen bestimmten Programmteil nur auf Anforderung durch einen Eingangsimpuls unabhängig vom Steuerungszyklus aufzurufen:

Der zeitkritische Teil des Anwendungsprogramms wird vom Anwender in einen Baustein vom Type PROGRAMM (PRG) zusammengefasst. Dieser Baustein wird zur Interrupt-Routine deklariert, indem einmalig (zur Initialisierungszeit) SET_INTERRUPT_I aufgerufen wird. Das hat zur Folge, dass dieser Programmteil immer dann ausgeführt wird, wenn eine Flanke am Eingang CHANNEL erkannt wird. Werden Ein- und Ausgänge in diesem Programmteil genutzt, werden diese ebenfalls in der Interrupt-Routine, ausgelöst durch die Eingangs-Flanke, gelesen oder beschrieben. Über die Eingänge READ_INPUTS, WRITE_OUTPUTS oder ANALOG_INPUTS kann das Lesen oder Schreiben unterbunden werden.

Innerhalb des Programmteils können also alle zeitkritischen Ereignisse bearbeitet werden, indem Eingänge oder globale Variablen verknüpft und Ausgänge beschrieben werden. So können auch Bausteine nur genau dann ausgeführt werden, wenn sie durch ein Eingangssignal angefordert werden.

HINWEIS

Damit der per Interrupt aufgerufene Programmteil nicht zusätzlich zyklisch aufgerufen wird, sollte er (mit Ausnahme des Initialisierungsaufrufes) im Zyklus übersprungen werden.

Der Eingang (CHANNEL), der zum Auslösen des Interrupt überwacht wird, kann in der Interrupt-Routine nicht initialisiert und weiter verarbeitet werden.

Die Laufzeit des Hauptzyklus plus die Summe der Laufzeiten aller per Interrupt aufgerufenen Programmteile muss stets innerhalb der max. zulässigen Zykluszeit bleiben!

Für die Datenkonsistenz zwischen Hauptprogramm und den im Interrupt laufenden Programmteilen ist der Anwender zuständig!

19866

Interrupt-Prioritäten:

- Alle per Interrupt aufgerufenen Programmteile haben die gleiche Priorität der Ausführung. Mehrere gleichzeitige Interrupts werden sequenziell in Reihenfolge ihres Auftretens abgearbeitet.
- Wird eine weitere Flanke am gleichen Eingang während der Ausführung des per Interrupt aufgerufenen Programmteils erkannt, wird dieser zur Bearbeitung eingetragen und das Programm nach Beendigung direkt wieder aufgerufen. Optional können durch Setzen des Glitch-Filters störende Mehrfachimpulse ausgefiltert werden.
- Das im Interupt laufende Programm kann durch höherpriorisierte Interrupts (z.B. CAN) unterbrochen werden.
- Belegen mehrere Interrupts den gleichen Kanal, erhält der zuletzt initialisierte FB (oder das PRG) den Kanal. Der zuvor definierte FB (oder das PRG) wird dann nicht mehr aufgerufen und liefert keine Daten mehr.

19365

HINWEIS

Die Eindeutigkeit der Ein- und Ausgänge im Zyklus wird durch die Interrupt-Routine aufgehoben. Deshalb wird nur ein Teil der Ein- und Ausgänge bedient. Wurden sie im Interrupt-Programm initialisiert, werden folgende Ein- und Ausgänge gelesen oder geschrieben:
• Eingänge: IN00...IN07
• Ausgänge: Q00...Q07

Auch globale Variablen verlieren ihre Eindeutigkeit, wenn auf sie quasi gleichzeitig im Zyklus und durch die Interrupt-Routine zugegriffen wird. Insbesondere größere Datentypen (z.B. DINT) sind von dieser Problematik betroffen.

Alle anderen Ein- und Ausgänge werden, wie üblich, einmalig im Zyklus bearbeitet.

Parameter der Eingänge

2383

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE (nur 1 Zyklus lang): Initialisierung des Bausteins FALSE: Baustein wird nicht ausgeführt
CHANNEL	BYTE	Nummer des Interrupt-Eingangs (0...7) 0...7 für die Eingänge IN0...IN7
MODE	BYTE	Art der Flanke am Eingang CHANNEL, die den Interrupt auslöst 1 = steigende Flanke (Standard-Wert) 2 = fallende Flanke 3 = steigende und fallende Flanke > 3 = Standard-Wert
READ_INPUTS	BOOL	TRUE: die Eingänge 0..7 vor Aufruf des Programms lesen und in die Eingangsmerker I00...I07 schreiben FALSE: nur den unter CHANNEL angegebenen Kanal lesen und in den dazugehörigen Eingangsmerker Ixx schreiben
WRITE_OUTPUTS	BOOL	TRUE: die aktuellen Werte der Ausgangsmerker Q00...Q07 nach Programmablauf auf die Ausgänge schreiben FALSE: keine Ausgänge schreiben
ANALOG_INPUTS	BOOL	TRUE: die Eingänge 0..7 lesen und die ungefilterten, unkalibrierten Analogwerte in die Merker ANALOG_IRQ00...07 schreiben FALSE: die Merker ANALOG_IRQ00...07 nicht schreiben

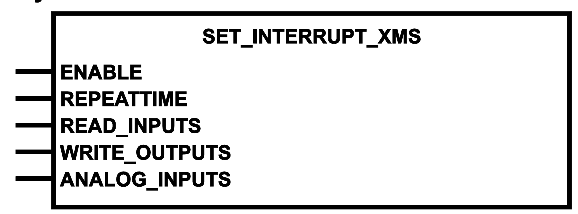
SET_INTERRUPT_XMS

272

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

275

SET_INTERRUPT_XMS organisiert das Ausführen eines Programmteils im Intervall von x ms.

In der klassischen SPS ist die Zykluszeit das Maß der Dinge für Echtzeitbetrachtungen. Gegenüber kundenspezifischen Steuerungen ist die SPS damit im Nachteil. Auch ein "Echtzeit-Betriebssystem" ändert nichts an dieser Tatsache, wenn das gesamte Anwendungsprogramm in einem einzigen unveränderlichen Block abläuft.

Ein möglicher Lösungsansatz wäre, die Zykluszeit kurz zu halten. Dieser Weg führt oft dazu, die Anwendung auf mehrere Steuerungszyklen zu verteilen. Die Programmierung wird dadurch jedoch unübersichtlich und schwierig.

Eine andere Möglichkeit besteht darin, einen bestimmten Programmteil in festen Zeitabständen (alle x ms) unabhängig vom Steuerungszyklus aufzurufen.

Der zeitkritische Teil des Anwendungsprogramms wird vom Anwender in einen Baustein vom Type PROGRAMM (PRG) zusammengefasst. Dieser Baustein wird zur Interrupt-Routine deklariert, indem einmalig (zur Initialisierungszeit) SET_INTERRUPT_XMS aufgerufen wird. Das hat zur Folge, dass dieser Programmteil immer nach Ablauf der REPEATTIME (alle x ms) abgearbeitet wird. Werden Ein- und Ausgänge in diesem Programmteil genutzt, werden diese ebenfalls im festgelegten Takt gelesen oder beschrieben. Über die Eingänge READ_INPUTS, WRITE_OUTPUTS oder ANALOG_INPUTS kann das Lesen oder Schreiben unterbunden werden.

Innerhalb des Programmteils können also alle zeitkritischen Ereignisse bearbeitet werden, indem Eingänge oder globale Variablen verknüpft und Ausgänge beschrieben werden. So können auch Zeitglieder genauer überwacht werden, als es in einem "normalen" Zyklus möglich ist.

> **HINWEIS**
>
> Damit der per Interrupt aufgerufene Programmteil nicht zusätzlich zyklisch aufgerufen wird, sollte er (mit Ausnahme des Initialisierungsaufrufes) im Zyklus übersprungen werden.
>
> Es können mehrere Timer-Interrupt-Bausteine aktiv sein. Der Zeitbedarf der Interrupt-Funktionen muss so berechnet werden, dass alle aufgerufenen Bausteine ausgeführt werden können. Das gilt besonders bei Berechnungen, Gleitkomma-Arithmetik und Regler-Funktionen.
>
> Für die Datenkonsistenz zwischen Hauptprogramm und den im Interrupt laufenden Programmteilen ist der Anwender zuständig!
>
> **Bitte beachten:** Bei einer hohen CAN-Busaktivität kann die eingestellte REPEATTIME schwanken.

971

HINWEIS

Die Eindeutigkeit der Ein- und Ausgänge im Zyklus wird durch die Interrupt-Routine aufgehoben. Deshalb wird nur ein Teil der Ein- und Ausgänge bedient. Wurden sie im Interrupt-Programm initialisiert, werden folgende Ein- und Ausgänge gelesen oder geschrieben.

Eingänge, digital:

%IX0.0...%IX0.7 (Controller: CR0n3n, CR7n3n)
%IX0.12...%IX0.15, %IX1.4...%IX1.8 (übrige ClassicController, ExtendedController, SafetyController)
%IX0.0, %IX0.8 (SmartController: CR250n)
IN08...IN11 (CabinetController: CR030n)
IN0...IN3 (Platinensteuerung: CS0015)

Eingänge, analog:

%IX0.0...%IX0.7 (Controller: CR0n3n, CR7n3n)
alle Kanäle (Auswahl bitcodiert) (alle übrigen Controller)

Ausgänge, digital:

%QX0.0...%QX0.7 (ClassicController, ExtendedController, SafetyController)
%QX0.0, %QX0.8 (SmartController: CR250n)
OUT00...OUT03 CabinetController: CR030n()
OUT0...OUT7 (Platinensteuerung: CS0015)

Auch globale Variablen verlieren ihre Eindeutigkeit, wenn auf sie quasi gleichzeitig im Zyklus und durch die Interrupt-Routine zugegriffen wird. Insbesondere größere Datentypen (z.B. DINT) sind von dieser Problematik betroffen.

Alle anderen Ein- und Ausgänge werden, wie üblich, einmalig im Zyklus bearbeitet.

Parameter der Eingänge

2382

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE (nur 1 Zyklus lang): Initialisierung des Bausteins FALSE: Baustein wird nicht ausgeführt
REPEATTIME	TIME	Zeitdauer in [ms] zwischen Ende des Programms und Neustart Die Zeitdauer zwischen zwei Aufrufen ermittelt sich damit als Summe aus REPEATTIME und Laufzeit des per Interrupt aufgerufenen Programms.
READ_INPUTS	BOOL	TRUE: die Eingänge 0..7 vor Aufruf des Programms lesen und in die Eingangsmerker I00...I07 schreiben FALSE: keine Aktualisierung der Eingänge
WRITE_OUTPUTS	BOOL	TRUE: die aktuellen Werte der Ausgangsmerker Q00...Q07 nach Programmablauf auf die Ausgänge schreiben FALSE: keine Ausgänge schreiben
ANALOG_INPUTS	BOOL	TRUE: die Eingänge 0..7 lesen und die ungefilterten, unkalibrierten Analogwerte in die Merker ANALOG_IRQ00...07 schreiben FALSE: die Merker ANALOG_IRQ00...07 nicht schreiben

5.2.8 Bausteine: Eingangswerte verarbeiten

Inhalt

1602
1302

Hier zeigen wir Ihnen **ifm**-Funktionsbausteine zum Lesen und Verarbeiten der analogen oder binären Signale am Geräte-Eingang.

HINWEIS

Die in der Steuerungskonfiguration von CODESYS erscheinenden analogen Rohwerte kommen direkt aus dem ADC. Sie sind noch nicht korrigiert!

Deshalb können in der Steuerungskonfiguration bei gleichen Geräten unterschiedliche Rohwerte erscheinen.

Erst durch die **ifm**-FBs findet eine Fehlerkorrektur und Normierung statt. Die FBs liefern den korrigierten Wert.

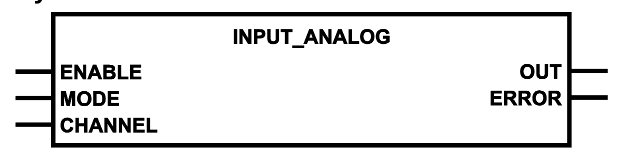
INPUT_ANALOG

15916

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

12912
12916

INPUT_ANALOG ermöglicht Strom- und Spannungsmessung an den Eingangskanälen (für alle Eingänge zugelassen).

Der FB liefert den aktuellen Analogwert am gewählten Analogkanal. Die Analogwerte werden normiert ausgegeben. Gleichzeitig werden die unkalibrierten Rohwerte über die Systemmerker ANALOGxx ausgegeben.

► Für Frequenz- und Periodenmessungen sowie Zählerfunktionen: MODE=1 (= IN_DIGITAL_H) einstellen!

Die Messung und der Ausgangswert resultieren aus der über MODE angegebenen Betriebsart:

MODE dez	MODE hex	Eingang Betriebsart	Ausgang OUT	Einheit
0	0000	deaktiviert	0	---
1	0001	Binäreingang minus-schaltend	0 / 1	---
2	0002	Binäreingang plus-schaltend	0 / 1	---
4	0004	Stromeingang	0…20 000	µA
8	0008	Spannungseingang	0…10 000	mV
16	0010	Spannungseingang	0…32 000	mV
32	0020	Spannungseingang ratiometrisch	0…1 000	‰
64	0040	---	---	---
128	0080	---	---	---
256	0100	---	---	---
512	0200	Widerstandseingang	3...680 16...30 000	Ω

Diese und weitere Betriebsarten der Eingänge: → FB *SET_INPUT_MODE* (→ Seite 132).



- Details → Kapitel *Mögliche Betriebsarten Ein-/Ausgänge* (→ Seite 232)
- bei VBBS < 4 V (für Modi 1, 2, 32) werden keine Werte eingelesen
- höhere Werte als angegeben werden ebenfalls erfasst (auch ratio)
- Überlastschutz ist bei Strommessung immer aktiv

18414

Falls Eingang I15 nicht verwendet:

► Eingang I15 als Binäreingang konfigurieren!

13020

HINWEIS

Nach dem Umschalten in einen anderen Modus während der Laufzeit dauert es wenige Zyklen, bis der Ausgangswert wieder korrekt ist.

Wenn derselbe Eingangskanal während der Laufzeit unterschiedlich konfiguriert wurde, dann gilt die zuletzt vorgenommene Konfiguration.

Parameter der Eingänge

11937

Parameter	Datentyp	Beschreibung		
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert		
MODE	WORD	Betriebsart des Eingangskanals CHANNEL:		
		0 = 0x0000	IN_NOMODE	(Aus; Voreinstellung aktiv)
		1 = 0x0001	IN_DIGITAL_H	Voreinstellung
		2 = 0x0002	IN_DIGITAL_L	
		4 = 0x0004	IN_CURRENT	0...20 000 µA
		8 = 0x0008	IN_VOLTAGE10	0...10 000 mV
		16 = 0x0010	IN_VOLTAGE30	0...30 000 mV
		32 = 0x0020	IN_RATIO	0...1 000 ‰
		64 = 0x0040	---	
		128 = 0x0080	---	
		256 = 0x0100	---	
		512 = 0x0200	IN_RESISTANCE	3...680 Ω / 16...30 000 Ω
CHANNEL	BYTE	Nummer des Eingangskanals (0...15) 0...15 für die Eingänge I00...I15 Für den FB xxx_E (falls vorhanden) gilt: 0...23 für die Eingänge I00_E...I23_E		

Parameter der Ausgänge

11938

Parameter	Datentyp	Beschreibung
OUT	WORD	Ausgangswert entsprechend MODE bei ungültiger Einstellung: OUT = "0"
ERROR	DWORD	Fehler-Code aus diesem FB-Aufruf → *Fehler-Codes* (→ Seite 306) (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für ERROR (n=beliebiger Wert):

Der 32-Bit-Fehler-Code besteht aus vier 8-Bit-Werten (DWORD).

4. Byte	3. Byte	2. Byte	1. Byte
Fehlerklasse	anwendungsspezifischer Fehler-Code	Fehlerquelle	Fehlerursache

Wert [hex]	Beschreibung
00 00 00 00	kein Fehler
01 00 10+nn 01	Kurzschluss am Eingangskanal Inn (nn = HEX-Wert) Beispiel: Kanal I15 ➩ 10+nn = 1F
01 00 10+nn 02	Unterbrechung am Eingangskanal Inn (nn = HEX-Wert) Beispiel: Kanal I15 ➩ 10+nn = 1F (nur bei IN_DIGITAL_H und DIAGNOSTICS = TRUE)
01 00 10+nn 04	Überlast am Eingangskanal Inn (nn = HEX-Wert) Beispiel: Kanal I15 ➩ 10+nn = 1F (nur bei Strommessung)
01 00 00 F8	falscher Parameter ➩ allgemeiner Fehler

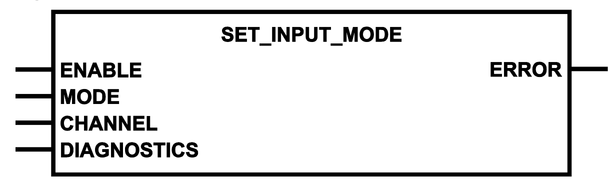
SET_INPUT_MODE

15918

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

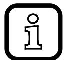
Symbol in CODESYS:



Beschreibung

11944

Mit SET_INPUT_MODE können Sie den Eingangskanälen Betriebsarten zuweisen.
→ Kapitel *Mögliche Betriebsarten Ein-/Ausgänge* (→ Seite 232)



- höhere Werte als angegeben werden ebenfalls erfasst (auch ratio)
- Überlastschutz ist bei Strommessung immer aktiv

Im laufenden Betrieb sollte die Betriebsart nicht geändert werden.

Auch mit dem FB *INPUT_ANALOG* (→ Seite 129) kann die Betriebsart an einem Eingang konfiguriert werden.

18414

Falls Eingang I15 nicht verwendet:
▶ Eingang I15 als Binäreingang konfigurieren!

13020

HINWEIS

Nach dem Umschalten in einen anderen Modus während der Laufzeit dauert es wenige Zyklen, bis der Ausgangswert wieder korrekt ist.

Wenn derselbe Eingangskanal während der Laufzeit unterschiedlich konfiguriert wurde, dann gilt die zuletzt vorgenommene Konfiguration.

Parameter der Eingänge

11945

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein initialisieren (nur 1 Zyklus) > Baustein-Eingänge lesen TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
MODE	WORD	Betriebsart des Eingangskanals CHANNEL: 0 = 0x0000 IN_NOMODE (Aus; Voreinstellung aktiv) 1 = 0x0001 IN_DIGITAL_H Voreinstellung 2 = 0x0002 IN_DIGITAL_L 4 = 0x0004 IN_CURRENT 0…20 000 µA 8 = 0x0008 IN_VOLTAGE10 0…10 000 mV 16 = 0x0010 IN_VOLTAGE30 0…30 000 mV 32 = 0x0020 IN_RATIO 0…1 000 ‰ 64 = 0x0040 --- 128 = 0x0080 ---
		256 = 0x0100 ---
		512 = 0x0200 IN_RESISTANCE 3…680 Ω / 16…30 000 Ω
CHANNEL	BYTE	Nummer des Eingangskanals (0...15) 0...15 für die Eingänge I00...I15 ℹ Für den FB xxx_E (falls vorhanden) gilt: 0...23 für die Eingänge I00_E...I23_E
DIAGNOSTICS	BOOL	TRUE: Kanal mit Diagnosefunktion nur wirksam für IN_DIGITAL_H oder IN_RESISTANCE Fehlermeldungen: • Leiterbruch oder Schluss gegen Masse bei Eingangsspannung < 1V für > 100 ms • Schluss gegen Versorgung bei Eingangsspannung > 95 % VBBS für > 100 ms • Widerstandswert > 31 kOhm (I12...I14) oder > 700 Ohm (I15) FALSE: Kanal ohne Diagnosefunktion

Parameter der Ausgänge

11947

Parameter	Datentyp	Beschreibung
ERROR	DWORD	Fehler-Code aus diesem FB-Aufruf → *Fehler-Codes* (→ Seite 306) (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für ERROR (n=beliebiger Wert):

Der 32-Bit-Fehler-Code besteht aus vier 8-Bit-Werten (DWORD).

4. Byte	3. Byte	2. Byte	1. Byte
Fehlerklasse	anwendungsspezifischer Fehler-Code	Fehlerquelle	Fehlerursache

Wert [hex]	Beschreibung
00 00 00 00	kein Fehler
01 00 00 F8	falscher Parameter ⇨ allgemeiner Fehler

5.2.9 Bausteine: analoge Werte anpassen

Inhalt

1603

Wenn die Werte analoger Eingänge oder die Ergebnisse von analogen Funktionen angepasst werden müssen, helfen Ihnen die folgenden Funktionsbausteine.

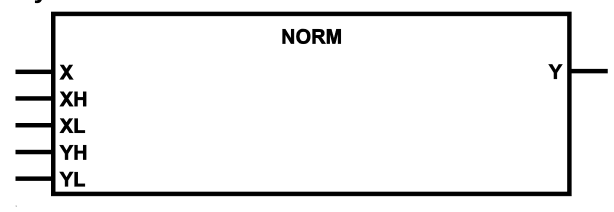
NORM

401

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

404

NORM normiert einen Wert innerhalb festgelegter Grenzen auf einen Wert mit neuen Grenzen.

Der FB normiert einen Wert vom Typ WORD, der innerhalb der Grenzen XH und XL liegt, auf einen Ausgangswert innerhalb der Grenzen YH und YL. Der FB wird z.B. bei der Erzeugung von PWM-Werten aus analogen Eingangsgrößen genutzt.

HINWEIS

► Der Eingangswert für X muss sich im definierten Bereich zwischen XL und XH befinden!
Der FB prüft NICHT den Wert X auf Plausibilität.

> Bedingt durch die Rundungsfehler können Abweichungen beim normierten Wert um 1 auftreten.

> Werden die Grenzen (XH/XL oder YH/YL) invertiert angegeben, erfolgt auch die Normierung invertiert.

Parameter der Eingänge

405

Parameter	Datentyp	Beschreibung
X	WORD	Eingangswert
XH	WORD	obere Grenze des Eingangswertebereichs [Inkremente]
XL	WORD	untere Grenze des Eingangswertebereichs [Inkremente]
YH	WORD	obere Grenze des Ausgangswertebereichs
YL	WORD	untere Grenze des Ausgangswertebereichs

Parameter der Ausgänge

406

Parameter	Datentyp	Beschreibung
Y	WORD	Ausgangswert

Beispiel: NORM (1)

407

unterer Grenzwert Eingang	0	XL
oberer Grenzwert Eingang	100	XH
unterer Grenzwert Ausgang	0	YL
oberer Grenzwert Ausgang	2000	YH

dann wandelt der Funktionsbaustein das Eingangssignal z.B. wie folgt um:

von X =	50	0	100	75
	↓	↓	↓	↓
nach Y =	1000	0	2000	1500

Beispiel: NORM (2)

408

unterer Grenzwert Eingang	2000	XL
oberer Grenzwert Eingang	0	XH
unterer Grenzwert Ausgang	0	YL
oberer Grenzwert Ausgang	100	YH

dann wandelt der Funktionsbaustein das Eingangssignal z.B. wie folgt um:

von X =	1000	0	2000	1500
	↓	↓	↓	↓
nach Y =	50	100	0	25

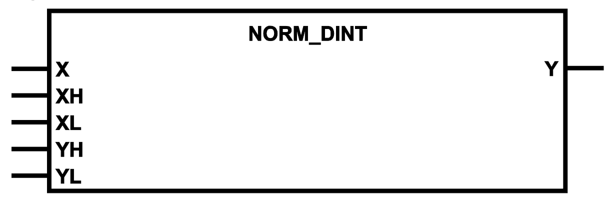
NORM_DINT

2217

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

2355

NORM_DINT normiert einen Wert innerhalb festgelegter Grenzen auf einen Wert mit neuen Grenzen.

Der FB normiert einen Wert vom Typ DINT, der innerhalb der Grenzen XH und XL liegt, auf einen Ausgangswert innerhalb der Grenzen YH und YL. Dieser FB wird z.B. bei der Erzeugung von PWM-Werten aus analogen Eingangsgrößen genutzt.

HINWEIS

- ► Der Eingangswert für X muss sich im definierten Bereich zwischen XL und XH befinden!
  Der FB prüft NICHT den Wert X auf Plausibilität.
- ► Das Ergebnis der Berechnung (XH-XL)•(YH-YL) muss im Wertebereich des Datentyps DINT (-2 147 483 648...2 147 483 647) bleiben!
- > Bedingt durch die Rundungsfehler können Abweichungen beim normierten Wert um 1 auftreten.
- > Werden die Grenzen (XH/XL oder YH/YL) invertiert angegeben, erfolgt auch die Normierung invertiert.

Parameter der Eingänge

2359

Parameter	Datentyp	Beschreibung
X	DINT	Eingangswert
XH	DINT	obere Grenze des Eingangswertebereichs
XL	DINT	untere Grenze des Eingangswertebereichs
YH	DINT	obere Grenze des Ausgangswertebereichs
YL	DINT	untere Grenze des Ausgangswertebereichs

Parameter der Ausgänge

2360

Parameter	Datentyp	Beschreibung
Y	DINT	Ausgangswert

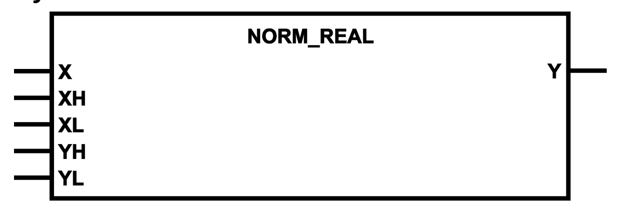
NORM_REAL

2218

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

2358

NORM_REAL normiert einen Wert innerhalb festgelegter Grenzen auf einen Wert mit neuen Grenzen.

Der FB normiert einen Wert vom Typ REAL, der innerhalb der Grenzen XH und XL liegt, auf einen Ausgangswert innerhalb der Grenzen YH und YL. Dieser FB wird z.B. bei der Erzeugung von PWM-Werten aus analogen Eingangsgrößen genutzt.

HINWEIS

- ► Der Eingangswert für X muss sich im definierten Bereich zwischen XL und XH befinden!
  Der FB prüft NICHT den Wert X auf Plausibilität.
- ► Das Ergebnis der Berechnung (XH-XL)•(YH-YL) muss im Wertebereich des Datentyps REAL (-3,402823466•10^{38}...3,402823466•10^{38}) bleiben!

> Bedingt durch die Rundungsfehler können Abweichungen beim normierten Wert um 1 auftreten.

> Werden die Grenzen (XH/XL oder YH/YL) invertiert angegeben, erfolgt auch die Normierung invertiert.

Parameter der Eingänge

2356

Parameter	Datentyp	Beschreibung
X	REAL	Eingangswert
XH	REAL	obere Grenze des Eingangswertebereichs
XL	REAL	untere Grenze des Eingangswertebereichs
YH	REAL	obere Grenze des Ausgangswertebereichs
YL	REAL	untere Grenze des Ausgangswertebereichs

Parameter der Ausgänge

2357

Parameter	Datentyp	Beschreibung
Y	REAL	Ausgangswert

5.2.10 Bausteine: Zählerfunktionen zur Frequenz- und Periodendauermessung

Inhalt

2322

Je nach Controller werden bis zu 16*) schnelle Eingänge unterstützt, die Eingangsfrequenzen bis zu 30 kHz verarbeiten können. Neben der reinen Frequenzmessung können die Eingänge auch zur Auswertung von inkrementellen Drehgebern (Zählerfunktion) eingesetzt werden.
*) ExtendedController: bis zu 32 schnelle Eingänge

Bedingt durch die unterschiedlichen Messmethoden können Fehler bei der Frequenzermittlung auftreten.

Zur einfachen Auswertung stehen folgende Bausteine zur Verfügung:

Baustein	zulässige Werte	Erklärung
FREQUENCY	0,1...30 000 Hz	Frequenz am angegebenen Kanal messen. Messfehler verringert sich bei hohen Frequenzen
PERIOD	0,1...5 000 Hz	Frequenz und Periodendauer (Zykluszeit) am angegebenen Kanal messen
PERIOD_RATIO	0,1...5 000 Hz	Frequenz und Periodendauer (Zykluszeit) sowie Puls-Pause-Verhältnis [‰] am angegebenen Kanal messen
FREQUENCY_PERIOD	0,1...30 000 Hz	Die Funktion vereinigt die beiden Funktionen FREQUENCY und PERIOD oder PERIOD_RATIO. Automatisches Umschalten der Messmethode bei 5 kHz
PHASE	0,1...5 000 Hz	Liest ein Kanalpaar ein und vergleicht die Phasenlage der Signale
INC_ENCODER	0,1...30 000 Hz	Vorwärts-/Rückwärts-Zählerfunktion zur Auswertung von Drehgebern
FAST_COUNT	0,1...30 000 Hz	Schnelle Impulse zählen

Wichtig bei Einsatz der schnellen Eingänge als "normale" Digitaleingänge:
- ► Die erhöhte Empfindlichkeit gegen Störimpulse beachten (z.B. Kontaktprellen bei mechanischen Kontakten).
- ► Das Eingangssignal bei Bedarf entprellen! → Kapitel *Hardware-Filter konfigurieren* (→ Seite 62)
- Der Standard-Digitaleingang kann Signale bis 50 Hz auswerten.

FAST_COUNT

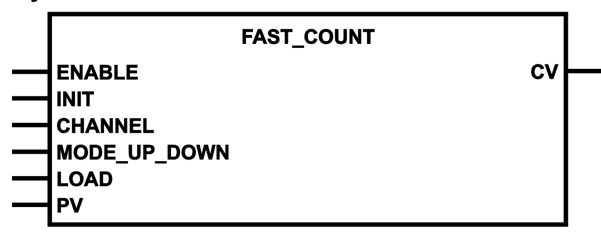
15922

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Für die Extended-Seite des ExtendedControllers endet der FB-Name mit "_E". (nicht bei CR0133)

Symbol in CODESYS:



Beschreibung

6830

FAST_COUNT arbeitet als Zählerbaustein für schnelle Eingangsimpulse.

Während ENABLE=TRUE erfasst der FB steigende Flanken an den FRQ-Eingangskanälen.
Maximale Eingangsfrequenz → Datenblatt.

Nach Rücksetzen und erneutem Setzen von ENABLE zählt der Zähler von dem Wert an weiter, der beim letzten Rücksetzen von ENABLE gültig war.

Mit Setzen von INIT (steigende Flanke) wird der Zählerwert CV=0 gesetzt.
Nach Rücksetzen des Parameters INIT zählt der Zähler von 0 an.

Am selben Eingang diesen FB **nicht** gemeinsam mit einem der folgenden FBs nutzen!

- *FREQUENCY* (→ Seite 142)
- *FREQUENCY_PERIOD* (→ Seite 144)
- *INC_ENCODER* (→ Seite 146)
- *PERIOD* (→ Seite 148)
- *PERIOD_RATIO* (→ Seite 150)
- *PHASE* (→ Seite 152)

Parameter der Eingänge

19869

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Zähler angehalten
INIT	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird initialisiert FALSE: im weiteren Programmablauf
CHANNEL	BYTE	Nummer des schnellen Eingangskanals (0...15) 0...11 für die Eingänge I00...I11 Für den FB xxx_E (falls vorhanden) gilt: 0...15 für die Eingänge I00_E...I15_E
MODE_UP_DOWN	BOOL	TRUE: Zähler zählt abwärts FALSE: Zähler zählt aufwärts
LOAD	BOOL	TRUE: Startwert PV wird in CV geladen FALSE: Funktion wird nicht ausgeführt
PV	DWORD	Startwert (Preset value) für den Zähler

Parameter der Ausgänge

572

Parameter	Datentyp	Beschreibung
CV	DWORD	aktueller Zählerwert Verhalten beim Überlauf: • zählt der Zähler abwärts, bleibt er bei 0 stehen • zählt der Zähler aufwärts, gibt es einen Überlauf.

FREQUENCY

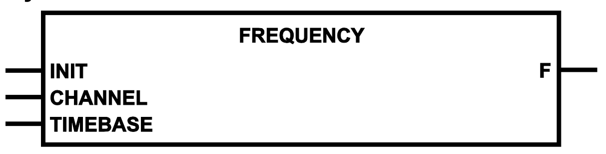
15924

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Für die Extended-Seite des ExtendedControllers endet der FB-Name mit "_E". (nicht bei CR0133)

Symbol in CODESYS:



Beschreibung

2325
20675

FREQUENCY misst die Frequenz des am gewählten Kanal (CHANNEL) ankommenden Signals. Der FB wertet dazu die positive Flanke des Signals aus.

In Abhängigkeit von der Zeitbasis (TIMEBASE) können Frequenzmessungen in einem weiten Wertebereich durchgeführt werden. Hohe Frequenzen erfordern eine kurze Zeitbasis, niedrige eine entsprechend längere. Grenzwerte:

TIMEBASE	zulässige, messbare Frequenz
57 000 ms (= Maximalwert)	1 149 Hz
2 184 ms	30 000 Hz (= Maximalwert)

Die Frequenz wird direkt in [Hz] ausgegeben.

Bei der Frequenzmessung sicherstellen, dass der FB innerhalb des Wertes von TIMEBASE nicht mehr als 65 535 positive Flanken empfängt!
Sonst kann das interne Zählregister überlaufen und zu falschen Ergebnissen führen.

Am selben Eingang diesen FB **nicht** gemeinsam mit einem der folgenden FBs nutzen!

- *FAST_COUNT* (→ Seite 140)
- *FREQUENCY_PERIOD* (→ Seite 144)
- *INC_ENCODER* (→ Seite 146)
- *PERIOD* (→ Seite 148)
- *PERIOD_RATIO* (→ Seite 150)
- *PHASE* (→ Seite 152)

Parameter der Eingänge

19871

Parameter	Datentyp	Beschreibung
INIT	BOOL	TRUE (nur 1 Zyklus lang): Baustein und Schnittstelle werden initialisiert FALSE: Messung läuft oder: Messung startet, wenn zuvor INIT=TRUE war
CHANNEL	BYTE	Nummer des schnellen Eingangskanals (0...15) 0...11 für die Eingänge I00...I11 Für den FB xxx_E (falls vorhanden) gilt: 0...15 für die Eingänge I00_E...I15_E
TIMEBASE	TIME	Zeitbasis zur Frequenzmessung (max. 57 s)

Parameter der Ausgänge

542

Parameter	Datentyp	Beschreibung
F	REAL	Frequenz des Eingangssignals in [Hz]

FREQUENCY_PERIOD

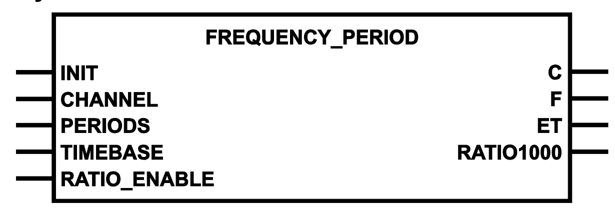
15926

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Für die Extended-Seite des ExtendedControllers endet der FB-Name mit "_E". (nicht bei CR0133)

Symbol in CODESYS:



Beschreibung

2335
20676

FREQUENCY_PERIOD misst die Frequenz und die Periodendauer (Zykluszeit) in [µs] am angegebenen Kanal (für alle Eingänge zugelassen). Maximale Eingangsfrequenz → Datenblatt.

Der FB vereinigt PERIOD oder PERIOD_RATIO und FREQUENCY in einem gemeinsamen Funktionsbaustein. Die Umschaltung der Messmethode erfolgt automatisch bei etwa 5 kHz:
- unterhalb von 5,2 kHz verhält sich der FB wie PERIOD oder PERIOD_RATIO
- oberhalb von 5,5 kHz verhält sich der FB wie FREQUENCY.

Der FB misst die Frequenz und die Zykluszeit des am gewählten Kanal (CHANNEL) anstehenden Signals. Zur Berechnung werden alle positiven Flanken ausgewertet und der Mittelwert über die Anzahl der angegebenen Perioden (PERIODS) gebildet.

Bei einer Eingangsfrequenz > 5 kHz und aktivem FREQUENCY-Modus kann der Ratio nicht gemessen werden.

Der maximale Messbereich beträgt ca. 15 min.

> Bei der Frequenzmessung sicherstellen, dass der FB innerhalb des Wertes von TIMEBASE nicht mehr als 65 535 positive Flanken empfängt!
> Sonst kann das interne Zählregister überlaufen und zu falschen Ergebnissen führen.

> **HINWEIS**
>
> Am selben Eingang diesen FB **nicht** gemeinsam mit einem der folgenden FBs nutzen!
> - *FAST_COUNT* (→ Seite 140)
> - *FREQUENCY* (→ Seite 142)
> - *INC_ENCODER* (→ Seite 146)
> - *PERIOD* (→ Seite 148)
> - *PERIOD_RATIO* (→ Seite 150)
> - *PHASE* (→ Seite 152)

Parameter der Eingänge

19872

Parameter	Datentyp	Beschreibung
INIT	BOOL	TRUE (nur 1 Zyklus lang): Baustein und Schnittstelle werden initialisiert FALSE: Messung läuft oder: Messung startet, wenn zuvor INIT=TRUE war
CHANNEL	BYTE	Nummer des schnellen Eingangskanals (0...15) 0...11 für die Eingänge I00...I11 Für den FB xxx_E (falls vorhanden) gilt: 0...15 für die Eingänge I00_E...I15_E
PERIODS	BYTE	Anzahl der Perioden, über die gemittelt wird (1...16) 0 : Ausgänge C und F werden nicht aktualisiert > 16 : wird auf 16 limitiert
TIMEBASE	TIME	Zeitbasis zur Frequenzmessung (max. 57 s)
RATIO_ENABLE	BOOL	TRUE: Ratio-Messung an RATIO1000 ausgeben FALSE: Ratio-Messung nicht ausgeben

Parameter der Ausgänge

2337

Parameter	Datentyp	Beschreibung
C	DWORD	Zykluszeit der erfassten Perioden in [µs] zulässig = 33...10 000 000 = 0x21...0x989680
F	REAL	Frequenz des Eingangssignals in [Hz]
ET	TIME	bei Messung der Periodendauer: (nutzbar bei sehr langsamen Signalen) RATIO_ENABLE = TRUE: Verstrichene Zeit seit dem letzten Flankenwechsel am Eingang RATIO_ENABLE = FALSE: Verstrichene Zeit seit der letzten positiven Flanke am Eingang bei anderen Messungen: ET = 0
RATIO1000	WORD	bei Messung der Periodendauer: Puls-/Periode-Verhältnis in [‰] Voraussetzungen: • Impulsdauer ≥ 100 µs • Frequenz < 5 kHz bei anderen Messungen: RATIO1000 = 0

INC_ENCODER

15928

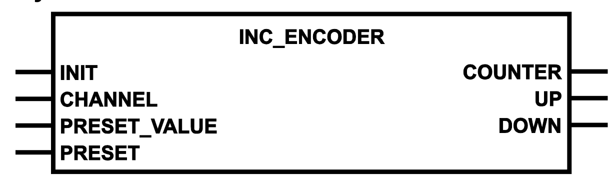
= Incremental Encoder

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Für die Extended-Seite des ExtendedControllers endet der FB-Name mit "_E". (nicht bei CR0133)

Symbol in CODESYS:



Beschreibung

2602

INC_ENCODER bietet eine Vorwärts-/Rückwärts-Zählerfunktion zur Auswertung von Drehgebern.

Immer zwei Frequenzeingänge bilden das Eingangspaar, das über den FB ausgewertet wird.

Grenzfrequenz = 30 kHz
max. anschließbar: 4 Drehgeber (ExtendedController: max. 8 Drehgeber)

Voreinstellwert setzen:
1. Wert in PRESET_VALUE eintragen
2. PRESET für einen Zyklus auf TRUE setzen
3. PRESET wieder auf FALSE setzen

Der FB zählt die Impulse an den Eingängen, solange INIT=FALSE und PRESET=FALSE sind.
Am Ausgang COUNTER steht der aktuelle Zählerstand an.

Die Ausgänge UP und DOWN zeigen die aktuelle Zählrichtung des Zählers an. Die Ausgänge sind dann TRUE, wenn im vorangegangenen Programmzyklus der Zähler in die entsprechende Richtung gezählt hat. Bleibt der Zähler stehen, wird auch der Richtungsausgang im folgenden Programmzyklus zurückgesetzt.

Am selben Eingang diesen FB **nicht** gemeinsam mit einem der folgenden FBs nutzen!
- *FAST_COUNT* (→ Seite 140)
- *FREQUENCY* (→ Seite 142)
- *FREQUENCY_PERIOD* (→ Seite 144)
- *PERIOD* (→ Seite 148)
- *PERIOD_RATIO* (→ Seite 150)
- *PHASE* (→ Seite 152)

Parameter der Eingänge

529

Parameter	Datentyp	Beschreibung
INIT	BOOL	TRUE (nur 1 Zyklus lang): Baustein wird initialisiert FALSE: im weiteren Programmablauf
CHANNEL	BYTE	Nummer des Eingangskanal-Paares (0...3) 0 = Kanalpaar 0 = Eingänge I00 + I01 ... 3 = Kanalpaar 3 = Eingänge I06 + I07 Für den FB xxx_E (falls vorhanden) gilt: 0 = Kanalpaar 0 = Eingänge I00_E + I01_E ... 3 = Kanalpaar 3 = Eingänge I06_E + I07_E
PRESET_VALUE	DINT	Zähler-Startwert
PRESET	BOOL	FALSE ⇨ TRUE (Flanke): PRESET_VALUE wird nach COUNTER geladen TRUE: Zähler ignoriert die Eingangsimpulse FALSE: Zähler zählt die Eingangsimpulse

Parameter der Ausgänge

530

Parameter	Datentyp	Beschreibung
COUNTER	DINT	aktueller Zählerstand
UP	BOOL	TRUE: Zähler zählte im letzten Zyklus aufwärts FALSE: Zähler zählte im letzten Zyklus nicht aufwärts
DOWN	BOOL	TRUE: Zähler zählte im letzten Zyklus abwärts FALSE: Zähler zählte im letzten Zyklus nicht abwärts

PERIOD

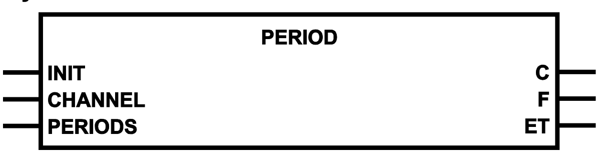
15930

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Für die Extended-Seite des ExtendedControllers endet der FB-Name mit "_E". (nicht bei CR0133)

Symbol in CODESYS:



Beschreibung

2330
20677

PERIOD misst die Frequenz und die Periodendauer (Zykluszeit) in [µs] am angegebenen Kanal (für alle Eingänge zugelassen). Maximale Eingangsfrequenz → Datenblatt.

Der FB misst die Frequenz und die Zykluszeit des am gewählten Kanal (CHANNEL) anstehenden Signals. Zur Berechnung werden alle positiven Flanken ausgewertet und der Mittelwert über die Anzahl der angegebenen Perioden (PERIODS) gebildet.

Bei niedrigen Frequenzen kommt es mit *FREQUENCY* (→ Seite 142) zu Ungenauigkeiten. Um dieses zu umgehen, kann PERIOD genutzt werden. Die Zykluszeit wird direkt in [µs] ausgegeben.

Der maximale Messbereich beträgt 10 Sekunden.

Am selben Eingang diesen FB **nicht** gemeinsam mit einem der folgenden FBs nutzen!

- *FAST_COUNT* (→ Seite 140)
- *FREQUENCY* (→ Seite 142)
- *FREQUENCY_PERIOD* (→ Seite 144)
- *INC_ENCODER* (→ Seite 146)
- *PERIOD_RATIO* (→ Seite 150)
- *PHASE* (→ Seite 152)

Parameter der Eingänge

19874

Parameter	Datentyp	Beschreibung
INIT	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird initialisiert FALSE: im weiteren Programmablauf
CHANNEL	BYTE	Nummer des schnellen Eingangskanals (0...15) 0...11 für die Eingänge I00...I11 Für den FB xxx_E (falls vorhanden) gilt: 0...15 für die Eingänge I00_E...I15_E
PERIODS	BYTE	Anzahl der Perioden, über die gemittelt wird (1...16) 0 : Ausgänge C und F werden nicht aktualisiert > 16 : wird auf 16 limitiert

Parameter der Ausgänge

375

Parameter	Datentyp	Beschreibung
C	DWORD	Zykluszeit der erfassten Perioden in [µs] zulässig = 200...10 000 000 = 0xC8...0x989680 (= 10 Sekunden)
F	REAL	Frequenz des Eingangssignals in [Hz]
ET	TIME	Verstrichene Zeit seit der letzten positiven Flanke am Eingang (nutzbar bei sehr langsamen Signalen)

PERIOD_RATIO

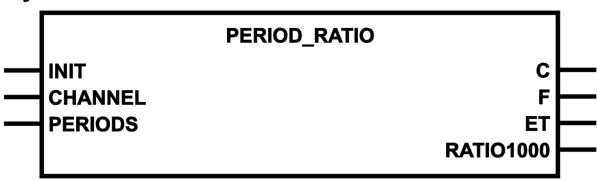
15932

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Für die Extended-Seite des ExtendedControllers endet der FB-Name mit "_E". (nicht bei CR0133)

Symbol in CODESYS:



Beschreibung

2332
20678

PERIOD_RATIO misst die Frequenz und die Periodendauer (Zykluszeit) in [µs] über die angegebenen Perioden am angegebenen Kanal (für alle Eingänge zugelassen). Zusätzlich wird das Puls-/Periode-Verhältnis in [‰] angegeben. Maximale Eingangsfrequenz → Datenblatt.

Dieser FB misst die Frequenz und die Zykluszeit des am gewählten Kanal (CHANNEL) anstehenden Signals. Zur Berechnung werden alle positiven Flanken ausgewertet und der Mittelwert über die Anzahl der angegebenen Perioden (PERIODS) gebildet. Zusätzlich wird das Puls-/Periode-Verhältnis in [‰] angegeben.
Zum Beispiel: Bei einem Signalverhältnis von 25 ms High-Pegel und 75 ms Low-Pegel, wird der Wert RATIO1000 von 250 ‰ ausgegeben.

Bei niedrigen Frequenzen kommt es mit *FREQUENCY* (→ Seite 142) zu Ungenauigkeiten. Um dieses zu umgehen, kann PERIOD_RATIO genutzt werden. Die Zykluszeit wird direkt in [µs] ausgegeben.

Der maximale Messbereich beträgt 10 Sekunden.

Am selben Eingang diesen FB **nicht** gemeinsam mit einem der folgenden FBs nutzen!
- *FAST_COUNT* (→ Seite 140)
- *FREQUENCY* (→ Seite 142)
- *FREQUENCY_PERIOD* (→ Seite 144)
- *INC_ENCODER* (→ Seite 146)
- *PERIOD* (→ Seite 148)
- *PHASE* (→ Seite 152)

Parameter der Eingänge

19873

Parameter	Datentyp	Beschreibung
INIT	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird initialisiert FALSE: im weiteren Programmablauf
CHANNEL	BYTE	Nummer des schnellen Eingangskanals (0...15) 0...11 für die Eingänge I00...I11 Für den FB xxx_E (falls vorhanden) gilt: 0...15 für die Eingänge I00_E...I15_E
PERIODS	BYTE	Anzahl der Perioden, über die gemittelt wird (1...16) 0 : Ausgänge C und F werden nicht aktualisiert > 16 : wird auf 16 limitiert

Parameter der Ausgänge

369

Parameter	Datentyp	Beschreibung
C	DWORD	Zykluszeit der erfassten Perioden in [µs] zulässig = 200...10 000 000 = 0xC8...0x989680 (= 10 Sekunden)
F	REAL	Frequenz des Eingangssignals in [Hz]
ET	TIME	Verstrichene Zeit seit dem letzten Zustandswechsel am Eingang (nutzbar bei sehr langsamen Signalen)
RATIO1000	WORD	bei Messung der Periodendauer: Puls-/Periode-Verhältnis in [‰] Voraussetzungen: • Impulsdauer ≥ 100 µs • Frequenz < 5 kHz bei anderen Messungen: RATIO1000 = 0

PHASE

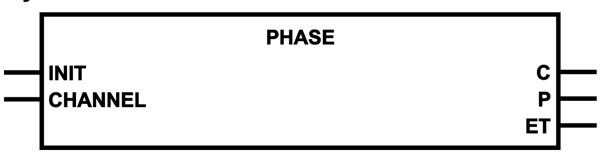
15934

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Für die Extended-Seite des ExtendedControllers endet der FB-Name mit "_E". (nicht bei CR0133)

Symbol in CODESYS:



Beschreibung

2338
20679

PHASE liest ein Kanalpaar mit schnellen Eingängen ein und vergleicht die Phasenlage der Signale. Maximale Eingangsfrequenz → Datenblatt.

Dieser FB fasst jeweils ein Kanalpaar mit schnellen Eingängen zusammen, so dass die Phasenlage zweier Signale zueinander ausgewertet werden kann. Es kann eine Periodendauer bis in den Sekundenbereich ausgewertet werden (max. 10 Sekunden).

Am selben Eingang diesen FB **nicht** gemeinsam mit einem der folgenden FBs nutzen!

- *FAST_COUNT* (→ Seite 140)
- *FREQUENCY* (→ Seite 142)
- *FREQUENCY_PERIOD* (→ Seite 144)
- *INC_ENCODER* (→ Seite 146)
- *PERIOD* (→ Seite 148)
- *PERIOD_RATIO* (→ Seite 150)

Parameter der Eingänge

19875

Parameter	Datentyp	Beschreibung
INIT	BOOL	TRUE (nur 1 Zyklus lang): Baustein und Schnittstelle werden initialisiert FALSE: im weiteren Programmablauf
CHANNEL	BYTE	Nummer des Eingangskanal-Paares (0...5) 0 = Kanalpaar 0 = Eingänge I00 + I01 ... 5 = Kanalpaar 5 = Eingänge I10 + I11 Für den FB xxx_E (falls vorhanden) gilt: 0 = Kanalpaar 0 = Eingänge I00_E + I01_E ... 7 = Kanalpaar 7 = Eingänge I14_E + I15_E

Parameter der Ausgänge

363

Parameter	Datentyp	Beschreibung
C	DWORD	Periodendauer des Signals am ersten Eingang des Kanalpaares in [µs]
P	INT	Winkel der Phasenverschiebung gültige Messung = 1...358 °
ET	TIME	Verstrichene Zeit seit der letzten positiven Flanke am zweiten Impulseingang des Kanalpaares

5.2.11 Bausteine: Ausgangsfunktionen allgemein

Inhalt

10462

Für dieses Gerät können Sie die Funktionsweise von einigen oder von allen Ausgängen einstellen.
Hier zeigen wir Ihnen geeignete Bausteine dazu.

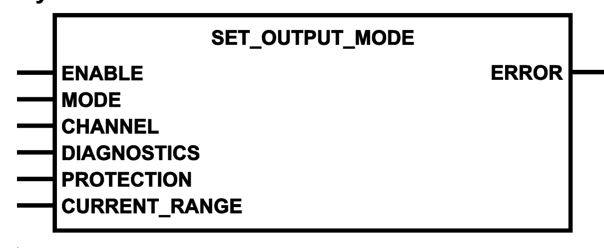
SET_OUTPUT_MODE

15937

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

12094

SET_OUTPUT_MODE setzt die Betriebsart des gewählten Ausgangskanals.

Zugelassene Betriebsarten (→ Datenblatt):

MODE	Konfig.-Wert	
	hex	dez
---	---	---
OUT_DIGITAL_H (plus)	0001	1
OUT_DIGITAL_L (minus)	0002	2

CURRENT_RANGE	Konfig.-Wert	
	hex	dez
keine Strommessung für MODE = 2	00	0
Strommessung 2 A (3 A) für MODE = 1	01	1
Strommessung 4 A für MODE = 1	02	2

Im laufenden Betrieb sollte die Betriebsart nicht geändert werden.

15672

HINWEIS

Soll im laufenden Betrieb am FB OUTPUT_BRIDGE der Messbereich für ACTUAL_CURRENT (auf 4 A) umgeschaltet werden?

► Für beide betreffenden Ausgänge den FB SET_OUTPUT_MODE in der Init-Phase **vor** dem FB OUTPUT_BRIDGE aufrufen!

► Beim Aufruf des FB SET_OUTPUT_MODE am FB OUTPUT_BRIDGE den Parameter DIRECTION berücksichtigen!
Das Umschalten des Messbereichs ist nur für den in B(H) betriebenen Ausgang zulässig, nämlich:

FB OUTPUT_BRIDGE: DIRECTION	H-Bridge	Messbereich für Ausgang ... umschalten
0	1 2	Q01(_E) Q09(_E)
1	1 2	Q03(_E) Q11(_E)

Parameter der Eingänge

12096

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein initialisieren (nur 1 Zyklus) > Baustein-Eingänge lesen TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
MODE	WORD	Betriebsart des Ausgangskanals CHANNEL: 1 = 0x0001 OUT_DIGITAL_H 2 = 0x0002 OUT_DIGITAL_L
CHANNEL	BYTE	Nummer des Ausgangskanals (0...15) 0...15 für die Ausgänge Q00...Q15 Für den FB xxx_E (falls vorhanden) gilt: 0...23 für die Ausgänge Q00_E...Q23_E
DIAGNOSTICS	BOOL	TRUE: Kanal mit Diagnosefunktion nur wirksam für OUT_DIGITAL_H • Leiterbruch bei Ausgängen mit Strommessung: wenn Strom < 25 mA für ≥ 66 ms • Leiterbruch bei Ausgängen ohne Strommessung: wenn Ausgangsspannung > 22 % VBBx für ≥ 66 ms • Überlast bei Strom > 112,5 % des Messbereichs für ≥ 66 ms • Kurzschluss bei Ausgangsspannung < 88 % VBBx für ≥ 66 ms FALSE: Kanal ohne Diagnosefunktion
PROTECTION	BOOL	TRUE: Schutz vor Überlast nur für OUT_DIGITAL_H UND Ausgang mit Strommessung Bei Erkennen von Überlast oder Kurzschluss schaltet der Ausgang für 1 s aus und dann wieder ein. FALSE: Funktion wird nicht ausgeführt
CURRENT_RANGE	BYTE	Strombegrenzung im Ausgangskanal CHANNEL: 0 = 0x00 CURRENT_RANGE_NONE (Aus) nur für Ausgang ohne Strommessung 1 = 0x01 CURRENT_RANGE1 2 A / 3 A nur für OUT_DIGITAL_H 2 = 0x02 CURRENT_RANGE2 4 A nur für OUT_DIGITAL_H

Parameter der Ausgänge

12102

Parameter	Datentyp	Beschreibung
ERROR	DWORD	Fehler-Code aus diesem FB-Aufruf → *Fehler-Codes* (→ Seite 306) (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für ERROR (n=beliebiger Wert):

Der 32-Bit-Fehler-Code besteht aus vier 8-Bit-Werten (DWORD).

4. Byte	3. Byte	2. Byte	1. Byte
Fehlerklasse	anwendungsspezifischer Fehler-Code	Fehlerquelle	Fehlerursache

Wert [hex]	Beschreibung
00 00 00 00	kein Fehler
01 00 00 F8	falscher Parameter ⇨ allgemeiner Fehler

5.2.12 Bausteine: PWM-Funktionen

Inhalt

13758

Hier finden Sie **ifm**-Bausteine, um die Ausgänge mit Pulsweitenmodulation (PWM) betreiben zu können.

OUTPUT_BRIDGE

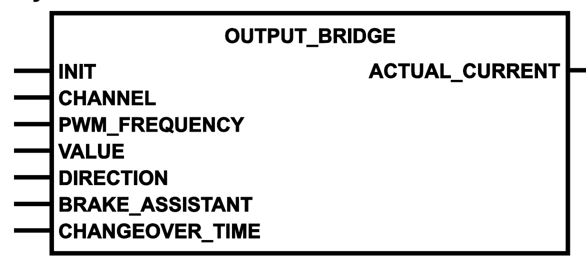
2198

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Für die Extended-Seite des ExtendedControllers endet der FB-Name mit "_E". (nicht bei CR0133)

Symbol in CODESYS:



Beschreibung

2203

OUTPUT_BRIDGE organisiert das Ansteuern der H-Brücken an den PWM-Kanälen.

Der FB dient zur einfachen Verwendung der Ausgänge als H-Brücke. Dazu werden jeweils zwei aufeinander folgende Ausgangskanäle mit minus-schaltendem Treiber zu einer Brücke zusammengefasst. Ist DIRECTION = FALSE, wird beim ersten Ausgang der plus-schaltende Treiber über ein PWM-Signal angesteuert und der minus-schaltende Treiber des zweiten Ausgangs ist durchgeschaltet.

HINWEIS

Bei Einsatz der H-Brücke wird die Stromregelung nicht unterstützt.

Ausgänge, die im PWM-Modus betrieben werden, unterstützen keine Diagnosefunktionen und es werden keine ERROR-Merker gesetzt.

Die Funktion OUT_OVERLOAD_PROTECTION ist in diesem Modus nicht aktiv!
Das Bit im Mode-Byte wird durch OUTPUT_BRIDGE zurückgesetzt.

Bei VALUE = 0 wird der Ausgang nicht komplett deaktiviert. Prinzipbedingt wird der Ausgang für die Dauer eines Timer-Ticks des PWM-Timers aktiv sein (typisch ca. 50 µs).

► FB in jedem SPS-Zyklus aufrufen!

Lage der als H-Brücke verwendbaren Ausgangskanäle → Datenblatt.

15672

HINWEIS

Soll im laufenden Betrieb am FB OUTPUT_BRIDGE der Messbereich für ACTUAL_CURRENT (auf 4 A) umgeschaltet werden?

► Für beide betreffenden Ausgänge den FB SET_OUTPUT_MODE in der Init-Phase **vor** dem FB OUTPUT_BRIDGE aufrufen!

► Beim Aufruf des FB SET_OUTPUT_MODE am FB OUTPUT_BRIDGE den Parameter DIRECTION berücksichtigen!
Das Umschalten des Messbereichs ist nur für den in B(H) betriebenen Ausgang zulässig, nämlich:

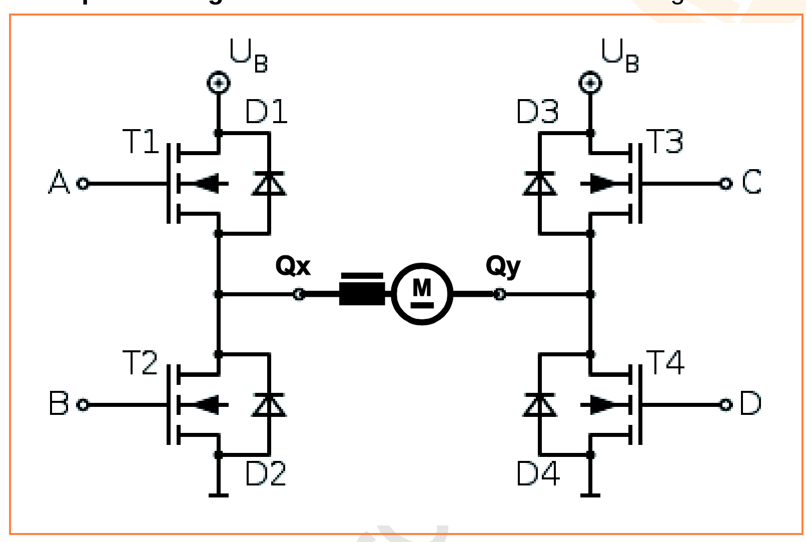
FB OUTPUT_BRIDGE: DIRECTION	H-Bridge	Messbereich für Ausgang ... umschalten
0	1 2	Q01(_E) Q09(_E)
1	1 2	Q03(_E) Q11(_E)

Prinzip der H-Brücke

9990
16411

Hier sehen Sie, wie eine H-Brücke am **ifm**-Controller via PWM-Ausgängen betrieben werden kann.

Prinzipschaltung einer H-Brücke mit PWM-Ansteuerung:



T1 und T2 bilden zusammen z.B. den Ausgang Qx.
Genauso bilden T3 und T4 z.B. den Ausgang Qy.
Dadurch werden nur zwei Anschlüsse für den DC-Motor benötigt.

Programm-Beispiel:

```
24 VAR
25     Init1: BOOL:=TRUE;
26     CycleTime:DWORD;
27     MaxCycleTime:DWORD;
28     ResetMax:BOOL;
29
30     DownloadID: CAN1_DOWNLOADID;
31
32 (*****************************************************************************
33 CHANNEL = 1:    Motor between OUT01 (Pin17) and OUT03(Pin15)
34 CHANNEL = 2:    Motor between OUT09 (Pin03) and OUT11(Pin05)
35 *****************************************************************************)
36     H_BRIDGE: OUTPUT_BRIDGE;
37     PWM_value: WORD := 100;          (* current PWM value - VALUE = 0...1000 *)
38     H_direction: BOOL;               (* TRUE = counter clockwise; FALSE = clockwise *)
39     H_current: WORD;                 (* output current in mA *)
40     changeover_time: WORD := 500;    (* Space time [ms] during which the motor is not triggered
41                                         (> 10 ms) in the case of a change of the rotational direction. *)
42 END_VAR
```

```
2
                        H_BRIDGE
                     OUTPUT_BRIDGE
       Init1-INIT             ACTUAL_CURRENT----H_current
           1-CHANNEL
         250-PWM_FREQUENCY
   PWM_value-VALUE
 H_direction-DIRECTION
       FALSE-BRAKE_ASSISTANT
        2000-CHANGEOVER_TIME
```

Parameter der Eingänge

2204

Parameter	Datentyp	Beschreibung
INIT	BOOL	TRUE (nur 1 Zyklus lang): Baustein wird initialisiert FALSE: im weiteren Programmablauf
CHANNEL	BYTE	Name des Ausgangspaares: 1 = Brücke 1 an Q01 + Q03 2 = Brücke 2 an Q09 + Q11 Für den FB xxx_E (falls vorhanden) gilt: 1 = Brücke 1 an Q01_E + Q03_E 2 = Brücke 2 an Q09_E + Q11_E
PWM_FREQUENCY	WORD	PWM-Frequenz [Hz] für die Last am Ausgang > FB begrenzt den Wert auf 20...2 000 = 0x0014...0x07D0
VALUE	WORD	PWM-Wert (Puls-Periode-Verhältnis) in [‰] zulässig = 0...1 000 = 0x0000...0x03E8 Werte > 1 000 gelten als = 1 000
DIRECTION	BOOL	Drehrichtung des Motors: TRUE: entgegen Uhrzeigersinn (ccw): Brücke 1: Stromfluss Q01(_E) ⇦ Q03(_E) Brücke 2: Stromfluss Q09(_E) ⇦ Q11(_E) FALSE: im Uhrzeigersinn (cw): Brücke 1: Stromfluss Q01(_E) ⇨ Q03(_E) Brücke 2: Stromfluss Q09(_E) ⇨ Q11(_E)
BRAKE_ASSISTANT	BOOL	TRUE: Beim Wechsel der Drehrichtung: FB schaltet beide Ausgänge gegen Masse, zwecks Bremswirkung am Motor, solange CHANGEOVER_TIME läuft. FALSE: Funktion wird nicht ausgeführt
CHANGEOVER_TIME	WORD	Pausezeit in [ms], während der bei einem Wechsel der Drehrichtung der Motor nicht angesteuert wird (≥ Zykluszeit, mindestens 10 ms) Werte < 10 ms gelten als = 10 ms

Parameter der Ausgänge

2205

Parameter	Datentyp	Beschreibung
ACTUAL_CURRENT	WORD	Ausgangsstrom in [mA]

OUTPUT_CURRENT

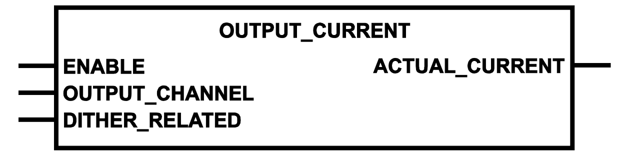
382

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Für die Extended-Seite des ExtendedControllers endet der FB-Name mit "_E".. (nicht bei CR0133)

Symbol in CODESYS:



Beschreibung

385

OUTPUT_CURRENT dient dem Messen des Stroms (optional: Mittelung über Dither-Periode) an einem Ausgangskanal.

Der FB liefert den aktuellen Ausgangsstrom, wenn die Ausgänge als PWM-Ausgänge oder als plus-schaltend benutzt werden. Die Strommessung erfolgt innerhalb des Gerätes, es werden also keine externen Messwiderstände benötigt.

Parameter der Eingänge

17894

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
OUTPUT_CHANNEL	BYTE	Nummer des stromgeregelten Ausgangskanals (0...15) 0...15 für die Ausgänge Q00...Q15 Für den FB xxx_E (falls vorhanden) gilt: 0...15 für die Ausgänge Q00_E...Q15_E
DITHER_RELATED	BOOL	Strom wird ermittelt als Mittelwert über... TRUE: eine Dither-Periode FALSE: eine PWM-Periode

Parameter der Ausgänge

387

Parameter	Datentyp	Beschreibung
ACTUAL_CURRENT	WORD	Ausgangsstrom in [mA]

OUTPUT_CURRENT_CONTROL

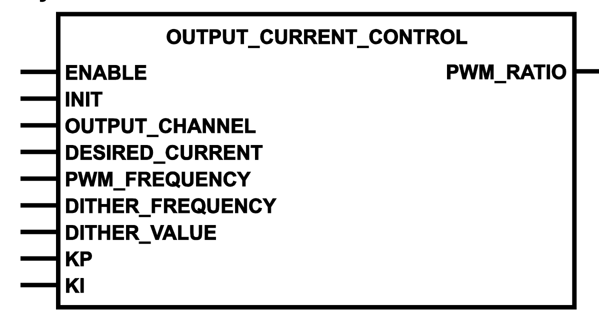
2196

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

ℹ Für die Extended-Seite des ExtendedControllers endet der FB-Name mit "_E". (nicht bei CR0133)

Symbol in CODESYS:



Beschreibung

2200

OUTPUT_CURRENT_CONTROL arbeitet als Stromregler für die PWM-Ausgänge.

Die beiden Anstellparameter KI und KP repräsentieren den I- und den P-Anteil des Reglers. Zur Ermittlung der besten Einstellung des Reglers bietet sich als Startwert an, KI = 50 und KP = 50 zu setzen. Je nach gewünschtem Reglerverhalten können die Werte schrittweise vergrößert (Regler wird härter / schneller) oder verkleinert (Regler wird schwächer / langsamer) werden.

Bei Sollwert DESIRED_CURRENT=0 wird der Ausgang innerhalb von etwa 100 ms auf 0 mA heruntergeregelt , wobei die Anstellparameter ignoriert werden.

HINWEIS

► Bei der Definition des Parameters DITHER_VALUE darauf achten, dass das resultierende PWM-Ratio im Arbeitsbereich der Regelung zwischen 0...1000 ‰ bleibt:
  • PWM-Ratio + DITHER_VALUE < 1000 ‰ und
  • PWM-Ratio - DITHER_VALUE > 0 ‰.
  Außerhalb dieses zulässigen Bereichs kann der im Parameter DESIRED_CURRENT angegebene Strom nicht erreicht werden.
  Außerhalb dieses zulässigen Bereichs wird DITHER_VALUE intern vorübergehend auf den maximal möglichen Wert reduziert, so dass der Mittelwert des PWM-Ratio dem geforderten Wert entspricht.

> Bei aktiviertem Dither werden Änderungen an PWM_FREQUENCY, DITHER_VALUE und DITHER_FREQUENCY erst nach Ende der aktuellen Dither-Periode angewendet.

> Kann der im Parameter DESIRED_CURRENT angegebene Strom nicht erreicht werden, weil das PWM-Ratioverhältnis schon bei 100 % ist, wird das durch die Systemvariable ERROR_CONTROL_Qx angezeigt.

> Bei KI = 0 findet keine Regelung statt.

> Ergibt sich bei der Regelung ein PWM_RATIO = 0, wird der Ausgang nicht komplett deaktiviert. Prinzipbedingt wird der Ausgang für die Dauer eines Timer-Ticks des PWM-Timers aktiv sein (typisch ca. 50 µs).

► Die Initialisierung des FBs (INIT=TRUE) darf nur einmalig für einen SPS Zyklus erfolgen.

► Der Aufruf dieses FB mit einem als B(L) konfigurierten Ausgang ist nicht zulässig.

► Ein als PWM-Ausgang definierter Ausgang kann anschließend nicht mehr als Binärausgang verwendet werden.

> Übersteigt der fließende Strom im eingeschalteten Zustand den Messbereich, kann keine Regelung mehr erfolgen, da der AD-Wandler am Messbereichsende ist und daher falsche Werte (den max. Wert) liefert.

Parameter der Eingänge

2201

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Regelung läuft weiter mit den zuletzt gültigen Parametern
INIT	BOOL	TRUE (nur 1 Zyklus lang): Baustein wird initialisiert FALSE: im weiteren Programmablauf
OUTPUT_CHANNEL	BYTE	Nummer des stromgeregelten Ausgangskanals (0...15) 0...15 für die Ausgänge Q00...Q15 Für den FB xxx_E (falls vorhanden) gilt: 0...15 für die Ausgänge Q00_E...Q15_E
DESIRED_CURRENT	WORD	Stromsollwert des Ausgangs in [mA] zulässig = 0...2 000 / 0...4 000 (abhängig vom Ausgang und der Konfiguration)
PWM_FREQUENCY	WORD	PWM-Frequenz [Hz] für die Last am Ausgang > FB begrenzt den Wert auf 20...2 000 = 0x0014...0x07D0
DITHER_FREQUENCY	WORD	Dither-Frequenz in [Hz] Wertebereich = 0...FREQUENCY / 2 FREQUENCY / DITHER_FREQUENCY muss geradzahlig sein! Alle anderen Werte erhöht der FB auf den nächst passenden Wert.
DITHER_VALUE	WORD	Spitze-Spitze-Wert des Dithers in [‰] zulässig = 0...1 000 = 0x0000...0x03E8
KP	BYTE	Proportional-Anteil des Ausgangsignals
KI	BYTE	Integral-Anteil des Ausgangsignals bei KI = 0 keine Regelung

Parameter der Ausgänge

2202

Parameter	Datentyp	Beschreibung
PWM_RATIO	WORD	Zu Kontrollzwecken: Anzeige PWM-Tastverhältnis 0...999 ‰

PWM1000

326

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Für die Extended-Seite des ExtendedControllers endet der FB-Name mit "_E". (nicht bei CR0133)

Symbol in CODESYS:

PWM1000
INIT
FREQUENCY
CHANNEL
VALUE
CHANGE
DITHER_VALUE
DITHER_FREQUENCY

Beschreibung

2311

PWM1000 initialisiert und parametriert einen PWM-fähigen Ausgang.

Der FB ermöglicht eine einfache Anwendung der PWM-Funktion im Gerät. Für jeden Kanal kann jeweils eine eigene PWM-Frequenz, das Puls-Periode-Verhältnis und der Dither eingestellt werden.

Die PWM-Frequenz FREQUENCY kann direkt in [Hz] und das Puls-Periode-Verhältnis VALUE in 1 ‰-Schritten angegeben werden.

Bei VALUE = 0 wird der Ausgang nicht komplett deaktiviert. Prinzipbedingt wird der Ausgang für die Dauer eines Timer-Ticks des PWM-Timers aktiv sein (typisch ca. 50 µs).

► Bei der Definition des Parameters DITHER_VALUE darauf achten, dass das resultierende PWM-Ratio im Arbeitsbereich der Regelung zwischen 0...1000 ‰ bleibt:
  • PWM-Ratio + DITHER_VALUE < 1000 ‰ und
  • PWM-Ratio - DITHER_VALUE > 0 ‰.
  Außerhalb dieses zulässigen Bereichs kann der im Parameter DESIRED_CURRENT angegebene Strom nicht erreicht werden.
  Außerhalb dieses zulässigen Bereichs wird DITHER_VALUE intern vorübergehend auf den maximal möglichen Wert reduziert, so dass der Mittelwert des PWM-Ratio dem geforderten Wert entspricht.

► Den FB dauerhaft aufrufen!

► Der Aufruf dieses FB mit einem als B(L) konfigurierten Ausgang ist nicht zulässig.

> **HINWEIS**
>
> Die Funktionsänderung eines als PWM-Funktion definierten Kanals im laufenden Betrieb ist nicht möglich. Die PWM-Funktion bleibt solange gesetzt, bis an der Steuerung ein Hardware-Reset durchgeführt wurde ⇨ Versorgungsspannung ausschalten und wieder einschalten.
>
> Bei hohen PWM-Frequenzen kann es systembedingt zu Differenzen kommen zwischen eingestelltem und ausgegebenem Ratio-Verhältnis.

Änderungen während der Laufzeit:

Immer, wenn Eingang CHANGE auf TRUE gesetzt ist, übernimmt der FB den Wert ...
• FREQUENCY nach der aktuellen PWM-Periode
• VALUE nach der aktuellen PWM-Periode
• DITHER_VALUE nach der aktuellen Dither-Periode
• DITHER_FREQUENCY nach der aktuellen Dither-Periode

Parameter der Eingänge

2312

Parameter	Datentyp	Beschreibung
INIT	BOOL	TRUE (nur 1 Zyklus lang): Baustein wird initialisiert Übernahme neuer Wert von FREQUENCY FALSE: im weiteren Programmablauf
FREQUENCY	WORD	PWM-Frequenz in [Hz] > FB begrenzt den Wert auf 20...2 000 = 0x0014...0x07D0
CHANNEL	BYTE	Nummer des PWM-Ausgangskanals (0...15) 0...15 für die Ausgänge Q00...Q15 Für den FB xxx_E (falls vorhanden) gilt: 0...15 für die Ausgänge Q00_E...Q15_E
VALUE	WORD	PWM-Wert (Puls-Periode-Verhältnis) in [‰] zulässig = 0...1 000 = 0x0000...0x03E8 Werte > 1 000 gelten als = 1 000
CHANGE	BOOL	TRUE: Übernahme neuer Wert von ... • FREQUENCY: nach der aktuellen PWM-Periode • VALUE: nach der aktuellen PWM-Periode • DITHER_VALUE: nach der aktuellen Dither-Periode • DITHER_FREQUENCY: nach der aktuellen Dither-Periode FALSE: geänderter PWM-Wert hat keinen Einfluss auf den Ausgang
DITHER_VALUE	WORD	Spitze-Spitze-Wert des Dithers in [‰] zulässig = 0...1 000 = 0x0000...0x03E8
DITHER_FREQUENCY	WORD	Dither-Frequenz in [Hz] Wertebereich = 0...FREQUENCY / 2 FREQUENCY / DITHER_FREQUENCY muss geradzahlig sein! Alle anderen Werte erhöht der FB auf den nächst passenden Wert.

5.2.13 Bausteine: Hydraulikregelung

Inhalt

13760

Die Bibliothek `ifm_HYDRAULIC_32bit_Vxxyyzz.Lib` enthält folgende Bausteine:

CONTROL_OCC (→ Seite 170)	OCC = Output Current Control (= stromgeregelter Ausgang) skaliert den Eingangswert [WORD] auf einen angegebenen Strombereich
JOYSTICK_0 (→ Seite 172)	skaliert Signale [INT] aus einem Joystick auf fest definierte Kennlinien, normiert auf 0…1000
JOYSTICK_1 (→ Seite 175)	skaliert Signale [INT] aus einem Joystick auf parametrierbare Kennlinien, normiert auf 0…1000
JOYSTICK_2 (→ Seite 179)	skaliert Signale [INT] aus einem Joystick auf einen parametrierbaren Kennlinien-Verlauf; die Normierung ist frei bestimmbar
NORM_HYDRAULIC (→ Seite 182)	normiert einen Wert [DINT] innerhalb festgelegter Grenzen auf einen Wert mit neuen Grenzen

Aus der Bibliothek `UTIL.Lib` (im CODESYS-Paket) werden folgende Bausteine benötigt:
- RAMP_INT
- CHARCURVE

Diese Bausteine werden von den FBs der Hydraulik-Bibliothek automatisch aufgerufen und parametriert.

Aus der Bibliothek `ifm_CR0033_Vxxyyzz.LIB` werden folgende Bausteine benötigt:

OUTPUT_CURRENT (→ Seite 163)	misst den Strom (Mittelung über Dither-Periode) an einem Ausgangskanal
OUTPUT_CURRENT_CONTROL (→ Seite 164)	Stromregler für einen PWMi-Ausgangskanal

Diese Bausteine werden von den FBs der Hydraulik-Bibliothek automatisch aufgerufen und parametriert.

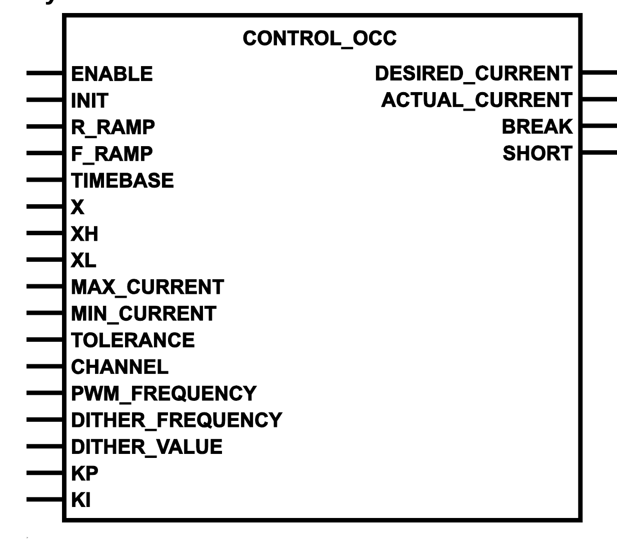
CONTROL_OCC

2735

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_HYDRAULIC_32bit_Vxxyyzz.Lib`

Symbol in CODESYS:



Beschreibung

2737

CONTROL_OCC skaliert den Eingangswert X auf einen angegebenen Strombereich.

Jede Instanz des FBs wird in jedem SPS-Zyklus einmalig aufgerufen.
Dieser FB nutzt aus der Bibliothek `ifm_CR0033_Vxxyyzz.LIB` folgende FBs:
- *OUTPUT_CURRENT_CONTROL* (→ Seite 164)
- *OUTPUT_CURRENT* (→ Seite 163)

Der Regler regelt in Abhängigkeit der Periodendauer des PWM Signals.

Die beiden Anstellparameter KI und KP repräsentieren den Integral- und den Proportionalanteil des Reglers. Zur Ermittlung der besten Einstellung des Reglers bietet sich als Startwert an, KI=50 und KP=50 zu setzen.

► Werte für KI und/oder KP vergrößern: ⇨ Regler wird schärfer / schneller
Werte für KI und/oder KP verkleinern: ⇨ Regler wird schwächer / langsamer

> Bei Ausgang DESIRED_CURRENT=0 wird der Ausgang **sofort** auf 0 mA geschaltet, wobei **nicht** entsprechend der eingestellten Parameter auf 0 mA heruntergeregelt wird.

Der Regler verfügt über einen schnellen Ausgleichsmechanismus bei Spannungseinbrüchen der Versorgungsspannung. In Abhängigkeit der Größe des Spannungseinbruchs wird zusätzlich zum Regelverhalten des Reglers die Ratio des PWMs dementsprechend so vergrößert, dass der Regler so schnell wie möglich den Sollwert erreicht.

ℹ Der Eingang X von CONTROL_OCC sollte von einem Ausgang der JOYSTICK-Bausteine gespeist werden.

Parameter der Eingänge

2739

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
INIT	BOOL	FALSE ⇨ TRUE (Flanke): Baustein wird initialisiert FALSE: im weiteren Programmablauf
R_RAMP	INT	Steigende Flanke der Rampe in [Inkremente/SPS-Zyklus] 0 = keine Rampe
F_RAMP	INT	Fallende Flanke der Rampe in [Inkremente/SPS-Zyklus] 0 = keine Rampe
TIMEBASE	TIME	Referenz für steigende und fallende Flanke der Rampe: t#0s = steigende / fallende Flanke in [Inkremente/SPS-Zyklus] Schnelle Controller haben sehr kurze Zykluszeiten! sonst = steigende / fallende Flanke in [Inkremente/TIMEBASE]
X	WORD	Eingangswert
XH	WORD	obere Grenze des Eingangswertebereichs [Inkremente]
XL	WORD	untere Grenze des Eingangswertebereichs [Inkremente]
MAX_CURRENT	WORD	Max. Ventilstrom in [mA]
MIN_CURRENT	WORD	Min. Ventilstrom in [mA]
TOLERANCE	BYTE	Toleranz für min. Ventilstrom in [Inkremente] Bei Überschreiten der Toleranz erfolgt Sprung auf MIN_CURRENT
CHANNEL	BYTE	Nummer des stromgeregelten Ausgangskanals (0...15) 0...15 für die Ausgänge Q00...Q15 Für den FB xxx_E (falls vorhanden) gilt: 0...15 für die Ausgänge Q00_E...Q15_E
PWM_FREQUENCY	WORD	PWM-Frequenz [Hz] für die Last am Ausgang
DITHER_FREQUENCY	WORD	Dither-Frequenz in [Hz] Wertebereich = 0...FREQUENCY / 2 FREQUENCY / DITHER_FREQUENCY muss geradzahlig sein! Alle anderen Werte erhöht der FB auf den nächst passenden Wert.
DITHER_VALUE	BYTE	Spitze-Spitze-Wert des Dithers in [%] zulässige Werte = 0...100 = 0x00...0x64
KP	BYTE	Proportional-Anteil des Ausgangsignals
KI	BYTE	Integral-Anteil des Ausgangsignals

Für KP, KI gilt: empfohlener Startwert = 50

Parameter der Ausgänge

602

Parameter	Datentyp	Beschreibung
DESIRED_CURRENT	WORD	Stromsollwert in [mA] für OCC (zu Kontrollzwecken)
ACTUAL_CURRENT	WORD	Ausgangsstrom in [mA]
BREAK	BOOL	Fehler: Leitung am Ausgang unterbrochen
SHORT	BOOL	Fehler: Kurzschluss in Leitung am Ausgang

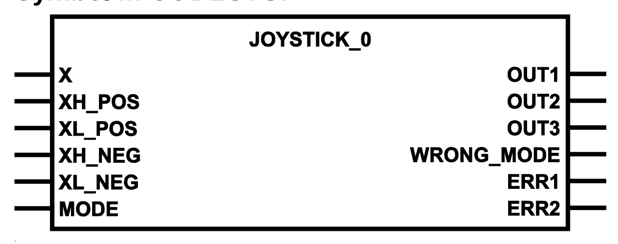
JOYSTICK_0

6250

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_hydraulic_32bit_Vxxyyzz.Lib`

Symbol in CODESYS:



Beschreibung

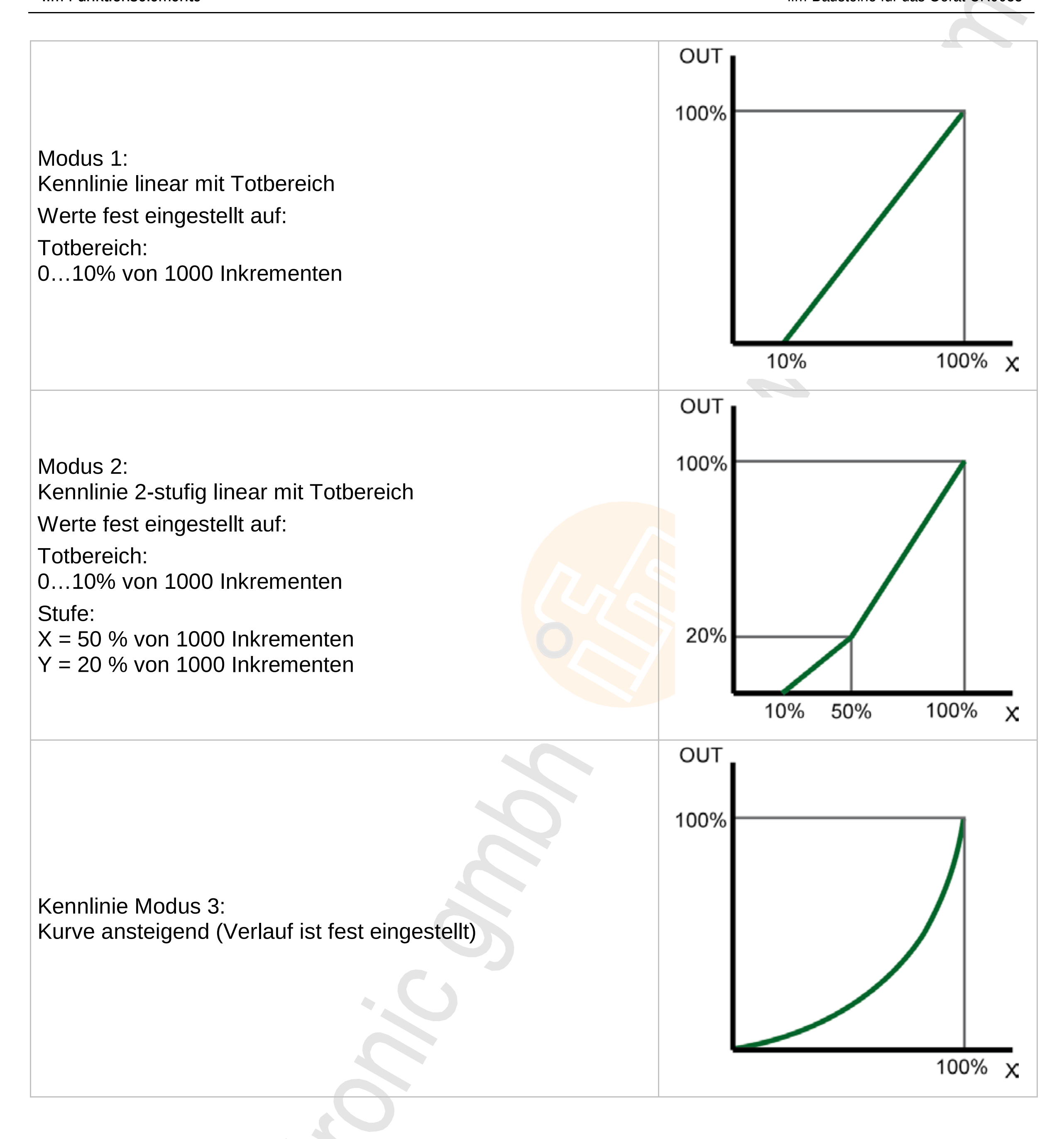
432

JOYSTICK_0 skaliert Signale aus einem Joystick auf fest definierte Kennlinien, normiert auf 0…1000.

Bei diesem FB sind die Kennlinien-Werte fest vorgegeben (→ Grafiken):

- Steigende Flanke der Rampe = 5 Inkremente/SPS-Zyklus
  ⓘ Schnelle Controller haben sehr kurze Zykluszeiten!
- Fallende Flanke der Rampe = keine Rampe

Die Parameter XL_POS (XL+), XH_POS (XH+), XL_NEG (XL-) und XH_NEG (XH-) dienen dazu, die Joystickbewegung nur im erwünschten Bewegungsbereich auszuwerten. Die Werte für den positiven und den negativen Bereich dürfen sich unterscheiden. Die Werte für XL_NEG und XH_NEG sind hier negativ.	XH- XL- XL+ XH+
Modus 0: Kennlinie linear für den Bereich XL bis XH	OUT 100% 100% X



Beschreibung	Kennlinie
Modus 1: Kennlinie linear mit Totbereich Werte fest eingestellt auf: Totbereich: 0…10% von 1000 Inkrementen	
Modus 2: Kennlinie 2-stufig linear mit Totbereich Werte fest eingestellt auf: Totbereich: 0…10% von 1000 Inkrementen Stufe: X = 50 % von 1000 Inkrementen Y = 20 % von 1000 Inkrementen	
Kennlinie Modus 3: Kurve ansteigend (Verlauf ist fest eingestellt)	

Parameter der Eingänge

433

Parameter	Datentyp	Beschreibung
X	INT	Eingangswert [Inkremente]
XH_POS	INT	Max. Sollwert positive Richtung [Inkremente] (auch negative Werte zulässig)
XL_POS	INT	Min. Sollwert positive Richtung [Inkremente] (auch negative Werte zulässig)
XH_NEG	INT	Max. Sollwert negative Richtung [Inkremente] (auch negative Werte zulässig)
XL_NEG	INT	Min. Sollwert negative Richtung [Inkremente] (auch negative Werte zulässig)
MODE	BYTE	Modus Auswahl Kennlinie: 0 = linear (X|OUT = 0|0 ... 1000|1000) 1 = linear mit Totbereich (X|OUT = 0|0 ... 100|0 ... 1000|1000) 2 = 2-stufig linear mit Totbereich (X|OUT = 0|0 ... 100|0 ... 500|200 ... 1000|1000) 3 = Kurve ansteigend (Verlauf ist fest eingestellt)

Parameter der Ausgänge

6252

Parameter	Datentyp	Beschreibung
OUT1	WORD	normierter Ausgangswert: 0…1000 Inkremente z.B. für Ventil links
OUT2	WORD	normierter Ausgangswert: 0…1000 Inkremente z.B. für Ventil rechts
OUT3	INT	normierter Ausgangswert: -1000…0…1000 Inkremente z.B. für Ventil an Ausgangsmodul (z.B. CR2011 oder CR2031)
WRONG_MODE	BOOL	Fehler: Ungültiger Modus
ERR1	BYTE	Fehler-Code für steigende Flanke (bezogen auf die intern verwendeten FBs CHARCURVE und RAMP_INT aus der `util.lib`) (mögliche Meldungen → folgende Tabelle)
ERR2	BYTE	Fehler-Code für fallende Flanke (bezogen auf die intern verwendeten FBs CHARCURVE und RAMP_INT aus der `util.lib`) (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für ERR1 und ERR2:

Wert dez	hex	Beschreibung
0	00	kein Fehler
1	01	Fehler in Zahlenreihe: Falsche Reihenfolge
2	02	Fehler: Eingangswert IN ist nicht im Wertebereich der Zahlenreihe
4	04	Fehler: Ungültige Anzahl N für Zahlenreihe

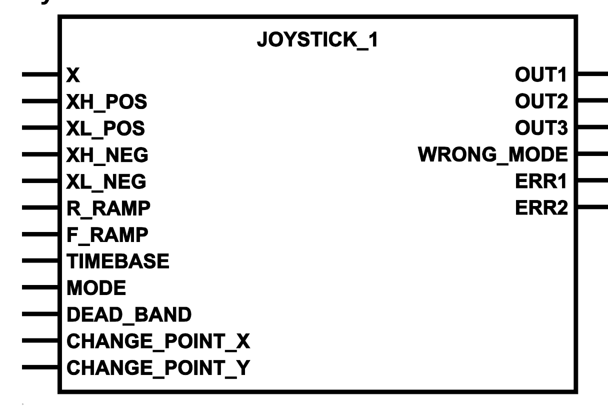
JOYSTICK_1

6255

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_hydraulic_32bit_Vxxyyzz.Lib`

Symbol in CODESYS:

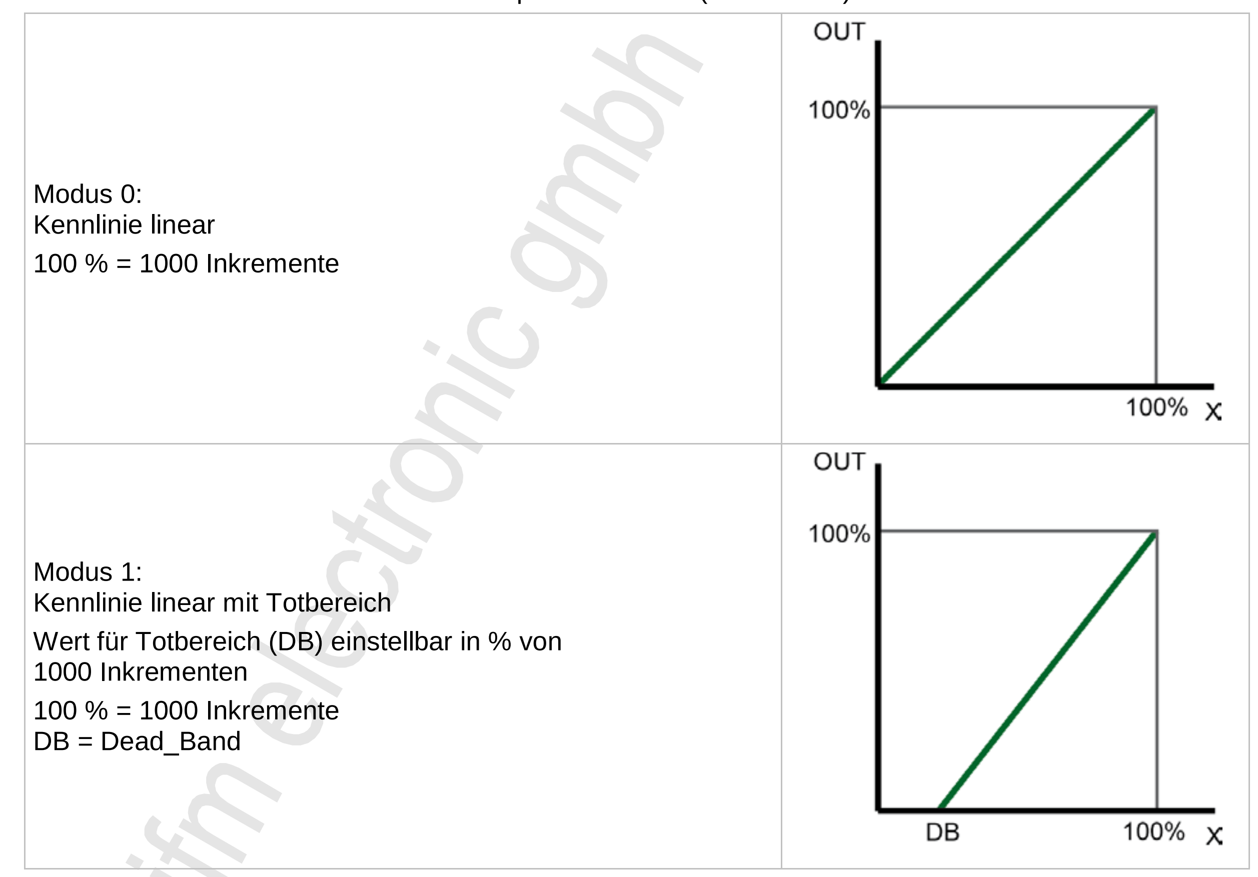


Beschreibung

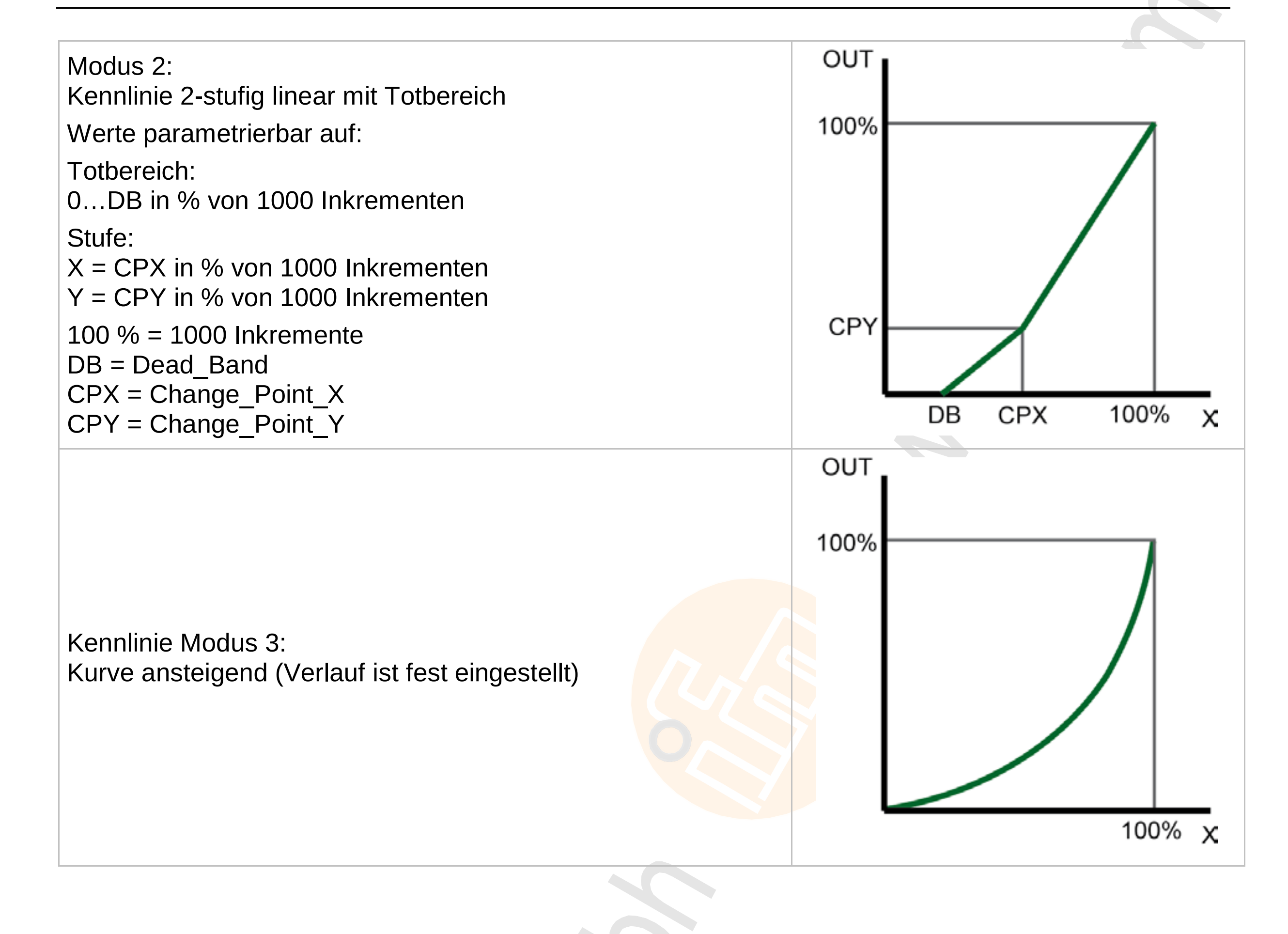
425

JOYSTICK_1 skaliert Signale aus einem Joystick auf parametrierbare Kennlinien, normiert auf 0…1000.

Bei diesem FB sind die Kennlinien-Werte parametrierbar (→ Grafiken):



Modus 0: Kennlinie linear 100 % = 1000 Inkremente	OUT 100% 100% X
Modus 1: Kennlinie linear mit Totbereich Wert für Totbereich (DB) einstellbar in % von 1000 Inkrementen 100 % = 1000 Inkremente DB = Dead_Band	OUT 100% DB 100% X



Modus 2: Kennlinie 2-stufig linear mit Totbereich Werte parametrierbar auf: Totbereich: 0…DB in % von 1000 Inkrementen Stufe: X = CPX in % von 1000 Inkrementen Y = CPY in % von 1000 Inkrementen 100 % = 1000 Inkremente DB = Dead_Band CPX = Change_Point_X CPY = Change_Point_Y	OUT 100% CPY DB CPX 100% X
Kennlinie Modus 3: Kurve ansteigend (Verlauf ist fest eingestellt)	OUT 100% 100% X

Parameter der Eingänge

6256

Parameter	Datentyp	Beschreibung
X	INT	Eingangswert [Inkremente]
XH_POS	INT	Max. Sollwert positive Richtung [Inkremente] (auch negative Werte zulässig)
XL_POS	INT	Min. Sollwert positive Richtung [Inkremente] (auch negative Werte zulässig)
XH_NEG	INT	Max. Sollwert negative Richtung [Inkremente] (auch negative Werte zulässig)
XL_NEG	INT	Min. Sollwert negative Richtung [Inkremente] (auch negative Werte zulässig)
R_RAMP	INT	Steigende Flanke der Rampe in [Inkremente/SPS-Zyklus] 0 = keine Rampe
F_RAMP	INT	Fallende Flanke der Rampe in [Inkremente/SPS-Zyklus] 0 = keine Rampe
TIMEBASE	TIME	Referenz für steigende und fallende Flanke der Rampe: t#0s = steigende / fallende Flanke in [Inkremente/SPS-Zyklus] ! Schnelle Controller haben sehr kurze Zykluszeiten! sonst = steigende / fallende Flanke in [Inkremente/TIMEBASE]
MODE	BYTE	Modus Auswahl Kennlinie: 0 = linear (X|OUT = 0|0 ... 1000|1000) 1 = linear mit Totbereich (X|OUT = 0|0 ... DB|0 ... 1000|1000) 2 = 2-stufig linear mit Totbereich (X|OUT = 0|0 ... DB|0 ... CPX|CPY ... 1000|1000) 3 = Kurve ansteigend (Verlauf ist fest eingestellt)
DEAD_BAND	BYTE	Einstellbarer Totbereich in [% von 1000 Inkrementen]
CHANGE_POINT_X	BYTE	Für Modus 2: Rampenstufe, Wert für X in [% von 1000 Inkrementen]
CHANGE_POINT_Y	BYTE	Für Modus 2: Rampenstufe, Wert für Y in [% von 1000 Inkrementen]

Parameter der Ausgänge

6252

Parameter	Datentyp	Beschreibung
OUT1	WORD	normierter Ausgangswert: 0…1000 Inkremente z.B. für Ventil links
OUT2	WORD	normierter Ausgangswert: 0…1000 Inkremente z.B. für Ventil rechts
OUT3	INT	normierter Ausgangswert: -1000…0…1000 Inkremente z.B. für Ventil an Ausgangsmodul (z.B. CR2011 oder CR2031)
WRONG_MODE	BOOL	Fehler: Ungültiger Modus
ERR1	BYTE	Fehler-Code für steigende Flanke (bezogen auf die intern verwendeten FBs CHARCURVE und RAMP_INT aus der `util.lib`) (mögliche Meldungen → folgende Tabelle)
ERR2	BYTE	Fehler-Code für fallende Flanke (bezogen auf die intern verwendeten FBs CHARCURVE und RAMP_INT aus der `util.lib`) (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für ERR1 und ERR2:

Wert dez	hex	Beschreibung
0	00	kein Fehler
1	01	Fehler in Zahlenreihe: Falsche Reihenfolge
2	02	Fehler: Eingangswert IN ist nicht im Wertebereich der Zahlenreihe
4	04	Fehler: Ungültige Anzahl N für Zahlenreihe

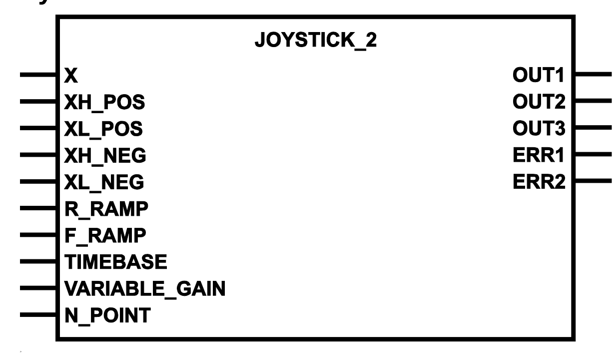
JOYSTICK_2

6258

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_hydraulic_32bit_Vxxyyzz.Lib`

Symbol in CODESYS:

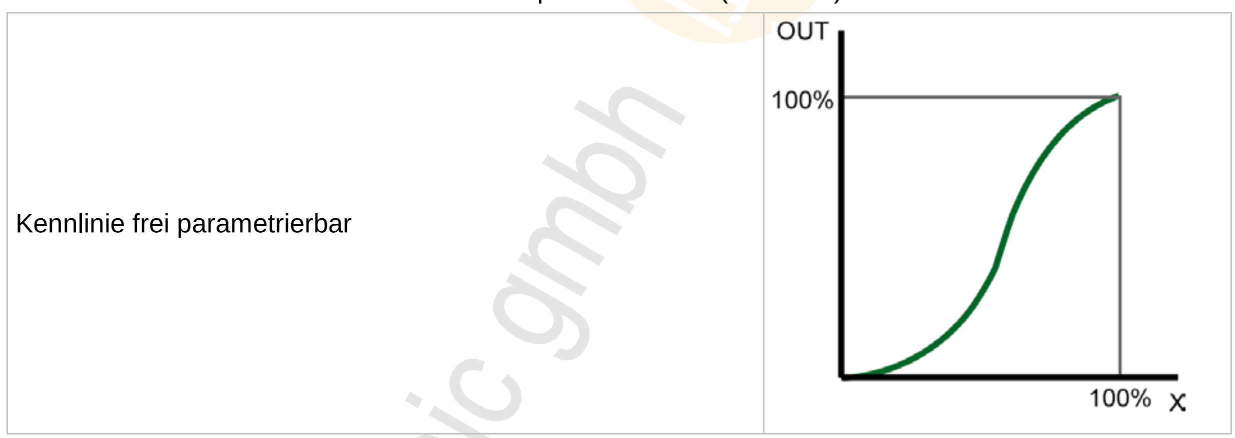


Beschreibung

418

JOYSTICK_2 skaliert Signale aus einem Joystick auf einen parametrierbaren Kennlinien-Verlauf. Die Normierung ist frei bestimmbar.

Bei diesem FB ist der Kennlinien-Verlauf frei parametrierbar (→ Grafik):



Parameter der Eingänge

6261

Parameter	Datentyp	Beschreibung
X	INT	Eingangswert [Inkremente]
XH_POS	INT	Max. Sollwert positive Richtung [Inkremente] (auch negative Werte zulässig)
XL_POS	INT	Min. Sollwert positive Richtung [Inkremente] (auch negative Werte zulässig)
XH_NEG	INT	Max. Sollwert negative Richtung [Inkremente] (auch negative Werte zulässig)
XL_NEG	INT	Min. Sollwert negative Richtung [Inkremente] (auch negative Werte zulässig)
R_RAMP	INT	Steigende Flanke der Rampe in [Inkremente/SPS-Zyklus] 0 = keine Rampe
F_RAMP	INT	Fallende Flanke der Rampe in [Inkremente/SPS-Zyklus] 0 = keine Rampe
TIMEBASE	TIME	Referenz für steigende und fallende Flanke der Rampe: t#0s = steigende / fallende Flanke in [Inkremente/SPS-Zyklus] (!) Schnelle Controller haben sehr kurze Zykluszeiten! sonst = steigende / fallende Flanke in [Inkremente/TIMEBASE]

VARIABLE_GAIN	ARRAY [0..10] OF POINT	Wertepaare, die den Kurven-Verlauf beschreiben
		Es werden die ersten in N_POINT angegebenen Wertepaare verwertet. n = 2…11
		Beispiel: 9 Wertepaare als Variable VALUES deklariert:
		`VALUES : ARRAY [0..10] OF POINT :=`
		`(X:=0,Y:=0),(X:=200,Y:=0), (X:=300,Y:=50),`
		`(X:=400,Y:=100), (X:=700,Y:=500),`
		`(X:=1000,Y:=900), (X:=1100,Y:=950),`
		`(X:=1200,Y:=1000), (X:=1400,Y:=1050);`
		Zwischen den Werten dürfen auch Leerzeichen stehen.

N_POINT	BYTE	Anzahl der Punkte (Wertepaare in VARIABLE_GAIN), womit die Kurven-Charakteristik definiert ist: n = 2…11

Parameter der Ausgänge

420

Parameter	Datentyp	Beschreibung
OUT1	WORD	normierter Ausgangswert: 0…1000 Inkremente z.B. für Ventil links
OUT2	WORD	normierter Ausgangswert: 0…1000 Inkremente z.B. für Ventil rechts
OUT3	INT	normierter Ausgangswert: -1000…0…1000 Inkremente z.B. für Ventil an Ausgangsmodul (z.B. CR2011 oder CR2031)
ERR1	BYTE	Fehler-Code für steigende Flanke (bezogen auf die intern verwendeten FBs CHARCURVE und RAMP_INT aus der `util.lib`) (mögliche Meldungen → folgende Tabelle)
ERR2	BYTE	Fehler-Code für fallende Flanke (bezogen auf die intern verwendeten FBs CHARCURVE und RAMP_INT aus der `util.lib`) (mögliche Meldungen → folgende Tabelle)

Mögliche Ergebnisse für ERR1 und ERR2:

Wert dez	hex	Beschreibung
0	00	kein Fehler
1	01	Fehler in Zahlenreihe: Falsche Reihenfolge
2	02	Fehler: Eingangswert IN ist nicht im Wertebereich der Zahlenreihe
4	04	Fehler: Ungültige Anzahl N für Zahlenreihe

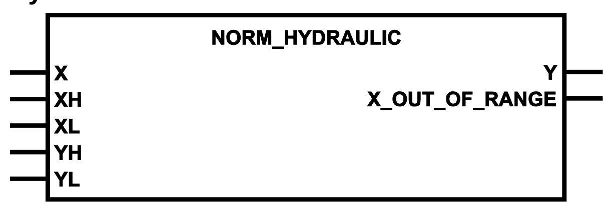
NORM_HYDRAULIC

394

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_hydraulic_32bit_Vxxyyzz.Lib`

Symbol in CODESYS:



Beschreibung

397

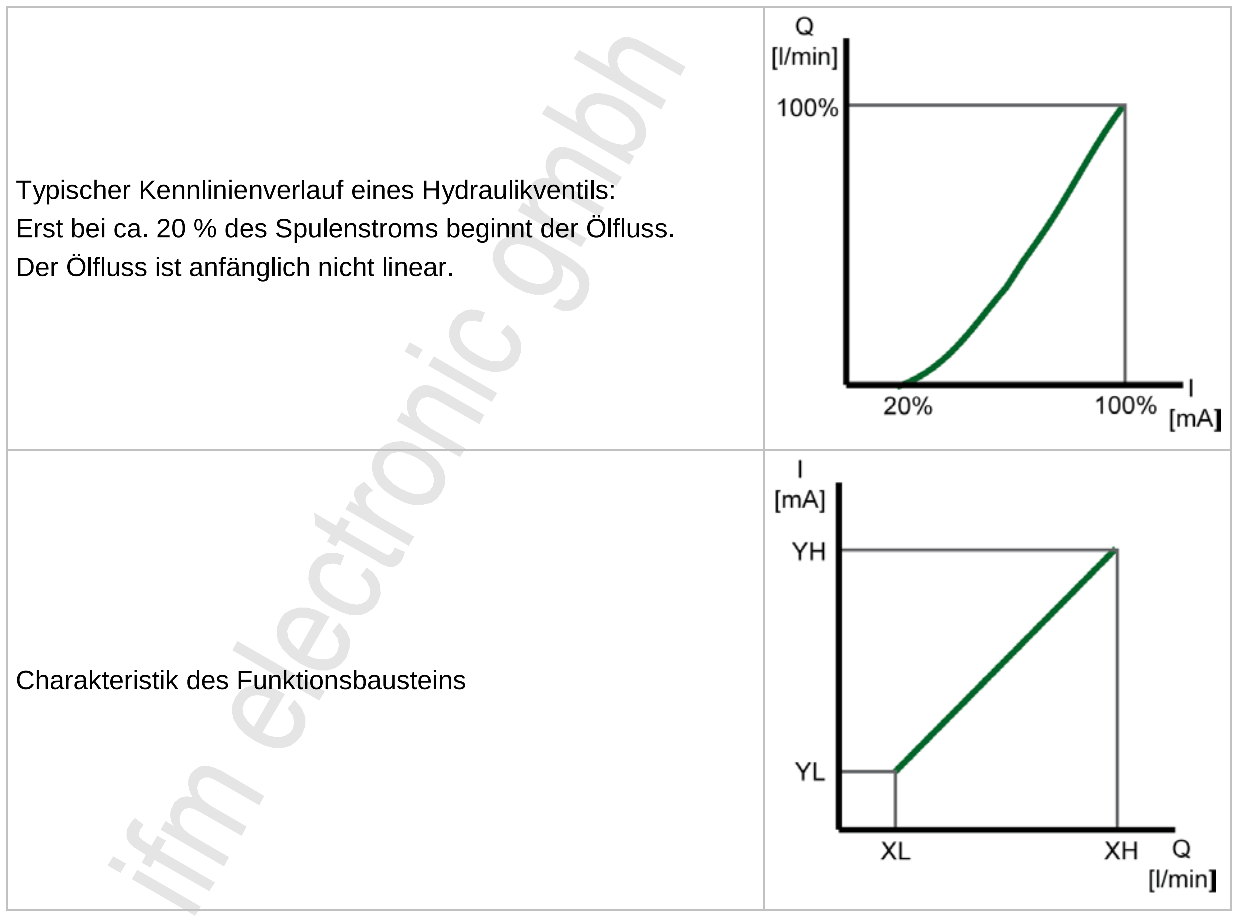
NORM_HYDRAULIC normiert Eingangswerte innerhalb festgesetzter Grenzen auf Werte mit neuen Grenzen.

Dieser FB entspricht NORM_DINT aus der CODESYS-Bibliothek `UTIL.Lib`.

Der FB normiert einen Wert vom Typ DINT, der innerhalb der Grenzen zwischen XH und XL liegt, auf einen Ausgangswert innerhalb der Grenzen zwischen YH und YL.

Bedingt durch Rundungsfehler können Abweichungen beim normierten Wert um 1 auftreten. Werden die Grenzen (XH/XL oder YH/YL) invertiert angegeben, erfolgt auch die Normierung invertiert.

Wenn X außerhalb der Grenzen XL…XH liegt, wird die Fehlermeldung X_OUT_OF_RANGE = TRUE.



Parameter der Eingänge

398

Parameter	Datentyp	Beschreibung
X	DINT	Eingangswert
XH	DINT	Max. Eingangswert [Inkremente]
XL	DINT	Min. Eingangswert [Inkremente]
YH	DINT	Max. Ausgangswert [Inkremente], z.B.: Ventilstrom [mA], Durchfluss [l/min]
YL	DINT	Min. Ausgangswert [Inkremente], z.B.: Ventilstrom [mA], Durchfluss [l/min]

Parameter der Ausgänge

399

Parameter	Datentyp	Beschreibung
Y	DINT	Ausgangswert
X_OUT_OF_RANGE	BOOL	Fehler: X liegt außerhalb der Grenzen von XH und XL

Beispiel: NORM_HYDRAULIC

400

Parameter	Fall 1	Fall 2	Fall 3
oberer Grenzwert Eingang XH	100	100	2000
unterer Grenzwert Eingang XL	0	0	0
oberer Grenzwert Ausgang YH	2000	0	100
unterer Grenzwert Ausgang YL	0	2000	0
nicht normierter Wert X	20	20	20
normierter Wert Y	400	1600	1

- Fall 1:
  Eingang mit relativ grober Auflösung.
  Ausgang mit hoher Auflösung.
  1 X-Inkrement ergibt 20 Y-Inkremente.
- Fall 2:
  Eingang mit relativ grober Auflösung.
  Ausgang mit hoher Auflösung.
  1 X-Inkrement ergibt 20 Y-Inkremente.
  Ausgangssignal ist gegenüber dem Eingangssignal invertiert.
- Fall 3:
  Eingang mit hoher Auflösung.
  Ausgang mit relativ grober Auflösung.
  20 X-Inkremente ergeben 1 Y-Inkrement.

5.2.14 Bausteine: Regler

Inhalt

1634

Der nachfolgende Abschnitt beschreibt im Detail die Bausteine, die zum Aufbau von Software-Reglern im **ecomat*mobile***-Gerät bereitgestellt werden. Die Bausteine können auch als Basis für die Entwicklung von eigenen Regelungsfunktionen genutzt werden.

Einstellregel für einen Regler

1627

Für Regelstrecken, deren Zeitkonstanten nicht bekannt sind, ist das Einstellverfahren nach Ziegler und Nickols im geschlossenen Regelkreis vorteilhaft:

Einstellregel

1628

Die Regeleinrichtung wird zunächst als eine reine P-Regeleinrichtung betrieben. Dazu wird die Vorhaltezeit T_V auf 0 und die Nachstellzeit T_N auf einen sehr großen Wert (ideal auf unendlich) für eine träge Strecke eingestellt. Bei einer schnellen Regelstrecke sollte ein kleines T_N gewählt werden.

Der Proportionalbeiwert KP wird anschließend solange vergrößert, bis die Regel- und die Stellabweichung bei KP = $KP_{kritisch}$ Dauerschwingungen mit konstanter Amplitude ausführen. Es ist damit die Stabilitätsgrenze erreicht.

Anschließend muss die Periodendauer $T_{kritisch}$ der Dauerschwingung ermittelt werden.

Nur bei Bedarf einen D-Anteil hinzufügen.

T_V sollte ca. 2...10-mal kleiner sein als T_N.

KP sollte gleich groß wie KD gewählt werden.

Idealisiert ist die Regelstrecke wie folgt einzustellen:

Regeleinrichtung	KP = KD	TN	TV
P	2,0 • $KP_{kritisch}$	—	—
PI	2,2 • $KP_{kritisch}$	0,83 • $T_{kritisch}$	—
PID	1,7 • $KP_{kritisch}$	0,50 • $T_{kritisch}$	0,125 • $T_{kritisch}$

! Bei diesem Einstellverfahren darauf achten, dass die Regelstrecke durch die auftretenden Schwingungen keinen Schaden nimmt. Bei empfindlichen Regelstrecken darf KP nur bis zu einem Wert erhöht werden, bei dem sicher noch keine Schwingungen auftreten.

Dämpfung von Überschwingungen

1629

Um Überschwingungen zu dämpfen, kann *PT1* (→ Seite 190) (Tiefpass) eingesetzt werden. Dazu wird der Sollwert XS durch das PT1-Glied gedämpft, bevor er der Reglerfunktion zugeführt wird.

Die Einstellgröße T1 sollte ca. 4...5-mal größer sein als TN des Reglers.

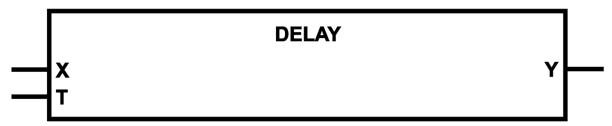
DELAY

585

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

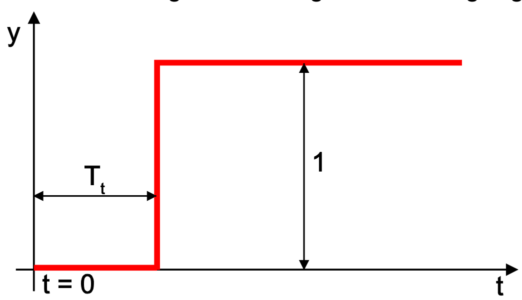
Symbol in CODESYS:



Beschreibung

588

DELAY verzögert die Ausgabe des Eingangswertes um die Zeit T (Totzeit-Glied).



Grafik: Zeitlicher Verlauf von DELAY

Die Totzeit wird durch die Dauer des SPS-Zyklus beeinflusst.
Die Totzeit darf nicht länger sein als 100 • SPS-Zykluszeit (Speichergrenze!).
Wird eine größere Verzögerung eingestellt, wird die Auflösung der Werte am Ausgang des FB schlechter, wodurch kurze Werteänderungen verloren gehen können.

Damit der FB einwandfrei arbeitet: FB in jedem SPS-Zyklus aufrufen!

Parameter der Eingänge

2615

Parameter	Datentyp	Beschreibung
X	REAL	Eingangswert
T	TIME	Verzögerungszeit (Totzeit) zulässig: 0...100 • Zykluszeit

Parameter der Ausgänge

2616

Parameter	Datentyp	Beschreibung
Y	REAL	Eingangswert, verzögert um die Zeit T

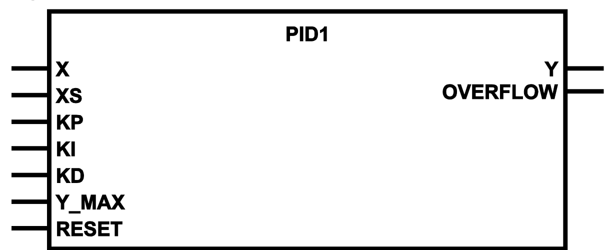
PID1

19235

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

19237

PID1 organisiert einen PID-Regler.

Die Änderung der Stellgröße eines PID-Reglers setzt sich aus einem proportionalen, integralen und differentialen Anteil zusammen.

Wenn der I-Anteil eine interne Begrenzung erreicht, weil eine Regelabweichung nicht ausgeregelt werden konnte, wird OVERFLOW = TRUE gemeldet.
OVERFLOW bleibt solange TRUE, solange die Begrenzung aktiv ist.

Parameter der Eingänge

19238

Parameter	Datentyp	Beschreibung
X	REAL	Eingangswert
XS	REAL	Sollwert
KP	REAL	Proportional-Anteil des Ausgangsignals (nur positive Werte zulässig)
KI	REAL	Integral-Anteil des Ausgangsignals (nur positive Werte zulässig)
KD	REAL	Differential-Anteil des Ausgangsignals (nur positive Werte zulässig)
Y_MAX	REAL	Maximaler Stellwert
RESET	BOOL	TRUE: Regler zurücksetzen FALSE: Funktion wird nicht ausgeführt

Parameter der Ausgänge

19241

Parameter	Datentyp	Beschreibung
Y	REAL	Ausgangswert
OVERFLOW	BOOL	TRUE: Überlauf des Datenpuffers ⇨ Datenverlust! FALSE: Datenpuffer ist ohne Datenverlust

Einstellempfehlung

19242

- Startwerte:
  KP = 0
  KD = 0
- KI dem Prozess anpassen.
- KP und KI anschließend schrittweise verändern.

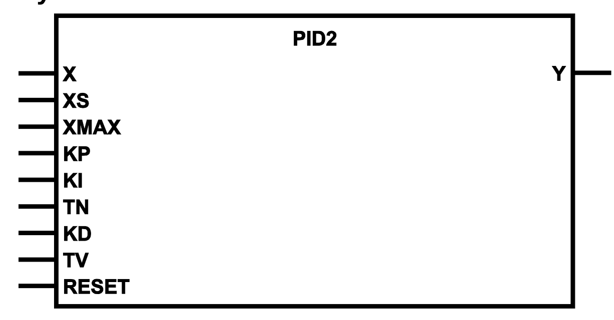
PID2

344

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

6262

PID2 organisiert einen PID-Regler.

Die Änderung der Stellgröße eines PID-Reglers setzt sich aus einem proportionalen, integralen und differentialen Anteil zusammen. Die Stellgröße ändert sich zunächst um einen von der Änderungsgeschwindigkeit der Eingangsgröße abhängigen Betrag (**D**ifferential-Anteil). Nach Ablauf der Vorhaltezeit TV geht die Stellgröße auf den dem Proportionalbereich entsprechenden Wert zurück und ändert sich dann entsprechend der Nachstellzeit TN.

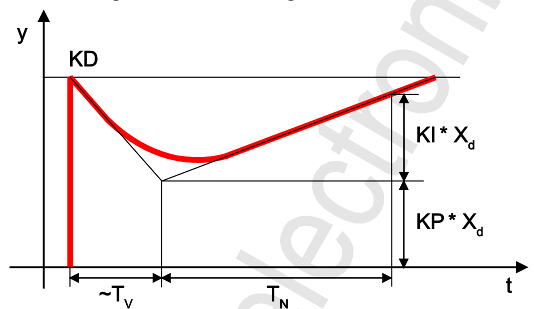
Die Stellgröße Y ist bereits auf *PWM1000* (→ Seite 167) normiert.

Regeln:

- Negative Werte bei KP, KI und KD sind nicht zulässig.
- Bei TN = 0 wird der I-Anteil nicht berechnet.
- Bei XS > XMAX wird XS auf XMAX limitiert.
- Bei X > XMAX wird Y auf 0 gesetzt.
- Wenn X > XS, dann wird die Stellgröße erhöht.
- Wenn X < XS, dann wird die Stellgröße reduziert.

Eine Führungsgröße wird intern zur Stellgröße hinzuaddiert:
Y = Y + 65 536 - (XS / XMAX • 65 536).

Die Stellgröße Y hat folgenden zeitlichen Verlauf.



Grafik: Typische Sprungantwort eines PID-Reglers

Parameter der Eingänge

12963

Parameter	Datentyp	Beschreibung
X	WORD	Eingangswert
XS	WORD	Sollwert
XMAX	WORD	Maximaler Istwert zur Festlegung des Istwert-Wertebereichs
KP	REAL	Proportional-Anteil des Ausgangsignals (nur positive Werte zulässig)
KI	REAL	Integral-Anteil des Ausgangsignals (nur positive Werte zulässig)
TN	TIME	Nachstellzeit (Integral-Anteil)
KD	REAL	Differential-Anteil des Ausgangsignals (nur positive Werte zulässig)
TV	TIME	Vorhaltezeit (Differential-Anteil)
RESET	BOOL	TRUE: Regler zurücksetzen FALSE: Funktion wird nicht ausgeführt

Parameter der Ausgänge

349

Parameter	Datentyp	Beschreibung
Y	WORD	Stellgröße (0...1000 ‰)

Einstellempfehlung

350

- ► TN gemäß des Zeitverhaltens der Strecke wählen
  (schnelle Strecke = kleines TN, träge Strecke = großes TN)
- ► KP langsam, schrittweise erhöhen bis zu einem Wert, bei dem sicher noch kein Schwingen auftritt.
- ► TN bei Bedarf nachjustieren
- ► Nur bei Bedarf D-Anteil hinzufügen:
  TV ca. 2...10-mal kleiner als TN wählen.
  KD etwa gleich groß wie KP wählen.

Beachten Sie, dass die maximale Regelabweichung + 127 beträgt. Für ein gutes Regelverhalten sollte dieser Bereich einerseits nicht überschritten, andererseits aber möglichst ausgenutzt werden.

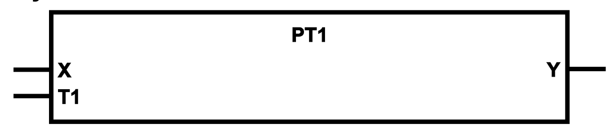
PT1

338

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

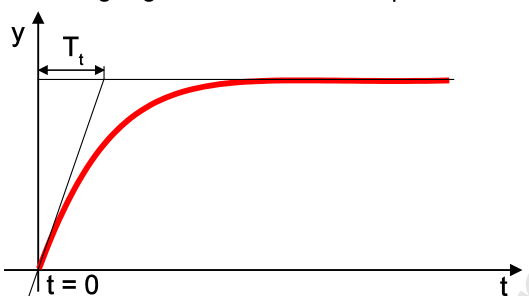
341

PT1 organisiert eine Regelstrecke mit Verzögerung 1. Ordnung.

Bei der Funktion handelt es sich um eine proportionale Regelstrecke mit Verzögerung. Sie wird z.B. zur Bildung von Rampen bei Einsatz der PWM-Funktionen genutzt.

> ⓘ Der Ausgang des FB kann instabil werden, wenn T1 kleiner ist als die SPS-Zykluszeit.

Die Ausgangsvariable Y des Tiefpassfilters hat folgenden zeitlichen Verlauf (Einheitssprungfunktion):



Grafik: Zeitlicher Verlauf bei PT1

Parameter der Eingänge

2618

Parameter	Datentyp	Beschreibung
X	DINT	Eingangswert
T1	TIME	Verzögerungszeit (Zeitkonstante)

Parameter der Ausgänge

2619

Parameter	Datentyp	Beschreibung
Y	DINT	Ausgangswert

5.2.15 Bausteine: Software-Reset

Inhalt

1594

Hiermit kann die Steuerung per Kommando im Anwendungsprogramm neu gestartet werden.

SOFTRESET

260

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

263

SOFTRESET führt einen kompletten Neustart des Geräts aus.

Die Funktion kann z.B. in Verbindung mit CANopen genutzt werden, wenn ein Node-Reset ausgeführt werden soll. Der FB SOFTRESET führt einen sofortigen Neustart der Steuerung durch. Der aktuelle Zyklus wird nicht beendet.

Vor dem Neustart erfolgt das Speichern der Retain- Variablen.
Der Neustart wird im Fehlerspeicher protokolliert.

> ❗ Bei einer laufenden Kommunikation: die lange Reset-Phase beachten, da andernfalls Guarding-Fehler gemeldet werden.

Parameter der Eingänge

264

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert

5.2.16 Bausteine: Zeit messen / setzen

Inhalt

1601

Mit folgenden Bausteinen der **ifm electronic** können Sie...

- Zeiten messen und im Anwendungsprogramm auswerten,
- bei Bedarf Zeitwerte ändern.

TIMER_READ

236

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

239

TIMER_READ liest die aktuelle Systemzeit aus.

Mit Anlegen der Versorgungsspannung bildet das Gerät einen Zeittakt, der in einem Register aufwärts gezählt wird. Dieses Register kann mittels des Funktionsaufrufes ausgelesen und z.B. zur Zeitmessung genutzt werden.

> ! Der System-Timer läuft maximal bis 0xFFFF FFFF (entspricht 49d 17h 2min 47s 295ms) und startet anschließend wieder mit 0.

Parameter der Ausgänge

241

Parameter	Datentyp	Beschreibung
T	TIME	Aktuelle Systemzeit [ms]

TIMER_READ_US

657

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

660

TIMER_READ_US liest die aktuelle Systemzeit in [µs] aus.

Mit Anlegen der Versorgungsspannung bildet das Gerät einen Zeittakt, der in einem Register aufwärts gezählt wird. Dieses Register kann mittels des FB-Aufrufes ausgelesen werden und z.B. zur Zeitmessung genutzt werden.

> **Info**
>
> Der System-Timer läuft maximal bis zum Zählerwert 1h 11min 34s 967ms 295µs und startet anschließend wieder mit 0.

Parameter der Ausgänge

662

Parameter	Datentyp	Beschreibung
TIME_US	DWORD	Aktuelle Systemzeit [µs]

5.2.17 Bausteine: Gerätetemperatur auslesen

Inhalt

2364

Mit folgendem Baustein zeigt Ihnen das Gerät die Innentemperatur.

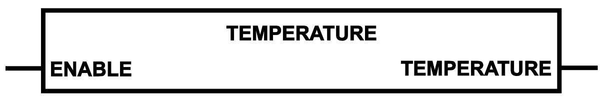
TEMPERATURE

2216

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

2365

TEMPERATURE liest die aktuelle Temperatur im Gerät aus.

Der FB kann zyklisch aufgerufen werden und zeigt am Ausgang die aktuelle Gerätetemperatur an (-40...125 °C).

Parameter der Eingänge

2366

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert

Parameter der Ausgänge

2367

Parameter	Datentyp	Beschreibung
TEMPERATURE	INT	Aktuelle Geräteinnentemperatur [°C]

5.2.18 Bausteine: Daten im Speicher sichern, lesen und wandeln

Inhalt

13795

Speicherarten zur Datensicherung

13805

Das Gerät bietet folgende Speicher:

Flash-Speicher

13803

Eigenschaften:

- nichtflüchtiger Speicher
- relativ langsames und nur blockweises Schreiben
- vor dem erneuten Schreiben muss Speicherinhalt gelöscht werden
- schnelles Lesen
- begrenzte Schreib-/Lesehäufigkeit
- nur zum Speichern großer Datenmengen sinnvoll einsetzbar
- Daten sichern mit FLASHWRITE
- Daten lesen mit FLASHREAD

FRAM-Speicher

13802

FRAM steht hier allgemein für alle Arten von nichtflüchtigen, schnellen Speichern.

Eigenschaften:

- schnelles Schreiben und Lesen
- unbegrenzte Schreib-/Lesehäufigkeit
- beliebige Speicherbereiche wählbar
- Daten sichern mit FRAMWRITE
- Daten lesen mit FRAMREAD

Dateisystem

2690

Das Dateisystem koordiniert, wo im Speicher welche Informationen liegen. Die Größe des Dateisystems beträgt 128 kByte.

Die Dateinamen des Dateisystems sind begrenzt:
max. Länge für Controller: CR0n3n, CR7n3n: 15 Zeichen
max. Länge für alle anderen Geräte: 11 Zeichen

Verhalten des Dateisystems im Controller: CR0n3n, CR7n3n**:**

- Der Controller versucht immer, die Datei zu schreiben, auch wenn der gleiche Dateiname bereits existiert. Gegebenenfalls wird die Datei mehrfach gespeichert.Genutzt wird nur die aktuelle Datei. Über den Download (s.u.) wird diese Mehrfach-Ablage vermieden.
- Einzelne Dateien können nicht überschrieben oder gelöscht werden.
- Das Dateisystem wird bei jedem Download (Bootprojekt-Download oder RAM-Download) komplett gelöscht. Anschließend kann z.B. eine Symboldatei oder eine Projektdatei (Funktionen in CODESYS) geschrieben werden.
- Das Dateisystem wird ebenfalls bei einem [Reset (Ursprung)] (CODESYS-Funktion im Menü [Online]) gelöscht.

Automatische Datensicherung

Inhalt

14168
2347

Die **ecomat*mobile***-Geräte bieten die Möglichkeit, Daten (BOOL, BYTE, WORD, DWORD) remanent (= spannungsausfallsicher) im Speicher zu sichern. Voraussetzung ist, dass die Daten als RETAIN-Variablen angelegt wurden (→ CODESYS).

Man unterscheidet zwischen Variablen, die als RETAIN deklariert wurden, und Variablen im Merkerbereich, der als Block mit *MEMORY_RETAIN_PARAM* (→ Seite 201) als remanent konfiguriert werden kann.
Details → Kapitel *Variablen* (→ Seite 66)

Der Vorteil des automatischen Speicherns ist, dass auch bei einem plötzlichen Spannungsabfall oder einer Unterbrechung der Versorgungsspannung die aktuellen Werte der Daten erhalten bleiben (z.B. Zählerstände).

! Wenn Versorgungsspannung < 8 V, werden keine Retain-Daten mehr gesichert! In diesem Fall wird Merker RETAIN_WARNING = TRUE.

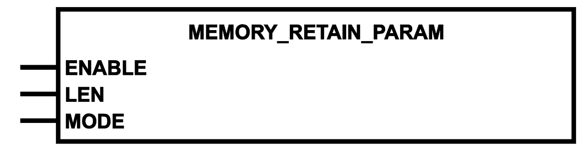
MEMORY_RETAIN_PARAM

2372

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

2374

MEMORY_RETAIN_PARAM legt das remanente Verhalten der Daten für verschiedene Ereignisse fest. In CODESYS als VAR_RETAIN deklarierte Variablen haben von vornherein ein remanentes Verhalten.

Remanente Daten behalten (wie die als VAR_RETAIN deklarierte Variablen) ihren Wert nach einem unkontrolliertem Beenden wie auch nach normalem Aus- und Einschalten der Steuerung. Bei erneutem Start arbeitet das Programm mit den gespeicherten Werten weiter.

Für (mit MODE) wählbare Gruppen von Ereignissen legt dieser FB fest, wie viele (LEN) Datenbytes (ab Merkerbyte %MB0) Retain-Verhalten haben sollen, auch wenn sie nicht ausdrücklich als VAR_RETAIN deklariert wurden.

Ereignis	MODE = 0	MODE = 1	MODE = 2	MODE = 3
Power OFF ⇨ ON	Daten werden neu initialisiert	Daten sind remanent	Daten sind remanent	Daten sind remanent
Reset warm	Daten werden neu initialisiert	Daten sind remanent	Daten sind remanent	Daten sind remanent
Reset kalt	Daten werden neu initialisiert	Daten werden neu initialisiert	Daten sind remanent	Daten sind remanent
Reset Ursprung	Daten werden neu initialisiert	Daten werden neu initialisiert	Daten sind remanent	Daten sind remanent
Anwendungsprogramm laden	Daten werden neu initialisiert	Daten werden neu initialisiert	Daten sind remanent	Daten sind remanent
Laufzeitsystem laden	Daten werden neu initialisiert	Daten werden neu initialisiert	Daten werden neu initialisiert	Daten sind remanent

Bei MODE = 0 habe nur solche Daten Retain-Verhalten wie bei MODE=1, die ausdrücklich als VAR_RETAIN deklariert wurden.

Wird der FB nie aufgerufen, verhalten sich die Merkerbytes nach MODE = 0. Die Merkerbytes, die oberhalb des konfigurierten Bereichs liegen, verhalten sich ebenfalls nach MODE = 0.

Eine einmal getätigte Konfiguration bleibt auf dem Gerät erhalten, auch wenn die Anwendung oder das Laufzeitsystem neu geladen werden.

Parameter der Eingänge

2375

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
LEN	WORD	Anzahl der Datenbytes ab Merkeradresse %MB0, die remanentes Verhalten haben sollen zulässig = 0...4 096 = 0x0...0x1000 LEN > 4 096 wird automatisch zu LEN = 4 096 korrigiert
MODE	BYTE	Ereignisse, bei denen diese Variablen Retain-Verhalten haben sollen (0...3; → Tabelle oben) Bei MODE > 3 bleibt die zuletzt gültige Einstellung erhalten

Manuelle Datensicherung

Inhalt

13801

Neben der Möglichkeit, die Daten automatisch zu sichern, können über FB-Aufrufe Anwenderdaten manuell in integrierte Speicher gesichert und von dort wieder gelesen werden.

Der Programmierer kann sich anhand der Speicheraufteilung (→ Kapitel *Verfügbarer Speicher* (→ Seite 14)) darüber informieren, welcher Speicherbereich frei zur Verfügung steht.

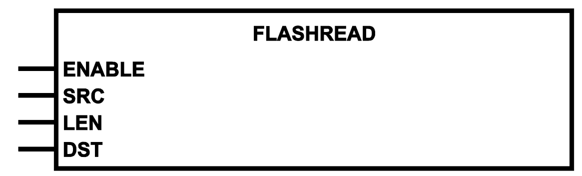
FLASHREAD

561

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

564

FLASHREAD ermöglicht das Lesen unterschiedlicher Datentypen direkt aus dem Flash-Speicher in den RAM.

> Der FB liest den Inhalt ab der Adresse von SRC aus dem Flash-Speicher. Dabei werden genau so viele Bytes übertragen, wie diese unter LEN angegeben sind.

> Das Lesen erfolgt komplett in dem Zyklus, in dem der FB aufgerufen wird.

► Darauf achten, dass der Zielspeicherbereich im RAM groß genug ist.

► Für die Zieladresse DST gilt:
(!) Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!

Parameter der Eingänge

2318

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
SRC	DWORD	Relative Quell-Anfangsadresse im Speicher zulässig = 0...65 535 = 0y0000 0000...0000 FFFF (!) Falls Startadresse außerhalb des zulässigen Bereichs: > kein Datentransfer
LEN	DWORD	Anzahl der Datenbytes (max. 65 536 = 0x0001 0000) (!) Würde durch die angegebene Anzahl an Bytes der Flash-Speicherbereich überschritten werden, werden die Daten nur bis zum Ende des Flash-Speicherbereichs übertragen.
DST	DWORD	Anfangsadresse der Zielvariablen (!) Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!

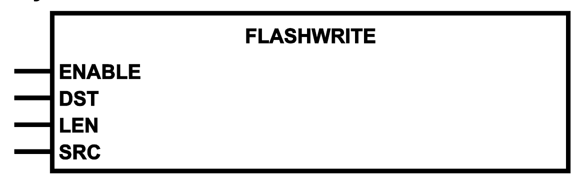
FLASHWRITE

555

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

19245

► Für den Einsatz des FBs den TEST-Eingang aktivieren! Ansonsten tritt ein Watchdog-Fehler auf.

Test-Eingang ist aktiv:
- Programmiermodus ist freigeben
- Software-Download ist möglich
- Zustand des Anwendungsprogramms ist abfragbar
- kein Schutz der gespeicherten Software möglich

558

⚠ WARNUNG

Gefahr durch unkontrollierten Prozessablauf!

Der Zustand der Ein-/Ausgänge wird während der Ausführung von FLASHWRITE "eingefroren".

► Diesen Funktionsbaustein nicht bei laufender Maschine ausführen!

FLASHWRITE ermöglicht das Schreiben unterschiedlicher Datentypen direkt in den Flash-Speicher.

Mit diesem FB sollen während der Inbetriebnahme große Datenmengen gesichert werden, auf die im Prozess nur lesend zugegriffen wird.

Der Flash-Speicher ist in 256 Byte große Pages organisiert.

► Wurde eine Page schon einmal (auch nur teilweise) beschrieben, muss der komplette Flash-Speicherbereich vor einem erneuten Schreibzugriff auf diese Page gelöscht werden. Dies geschieht durch einen Schreibzugriff auf die Adresse 0.

► Niemals mehrfach in eine Page schreiben! Erst immer alles löschen!
Sonst entstehen Traps oder Watchdog-Fehler.

► (!) Den Flash-Speicherbereich nicht öfter als 100mal löschen, da ansonsten die Datenkonsistenz in anderen Flash-Speicherbereichen nicht mehr gewährleistet werden kann.

► In jedem SPS-Zyklus darf FLASHWRITE nur einmalig gestartet werden!

► Für die Zieladresse DST gilt:
(!) Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!

> Der FB schreibt den Inhalt der Adresse SRC in den Flash-Speicher. Dabei werden genau so viele Bytes übertragen, wie diese unter LEN angegeben sind.

(!) Falls Startadresse SRC außerhalb des zulässigen Bereichs: kein Datentransfer!

Parameter der Eingänge

2603

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
DST	DWORD	Relative Ziel-Anfangsadresse im Speicher zulässig = 0...65 535 = 0x0...0x0000 FFFF
LEN	DWORD	Anzahl der Datenbytes (max. 65 536 = 0x0001 0000) Würde durch die angegebene Anzahl an Bytes der Flash-Speicherbereich überschritten werden, werden die Daten nur bis zum Ende des Flash-Speicherbereichs übertragen.
SRC	DWORD	Anfangsadresse der Quellvariablen Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!

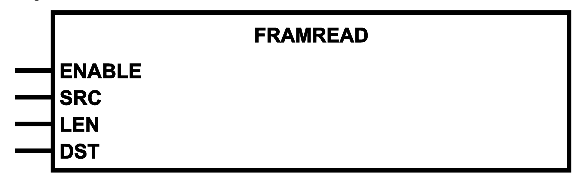
FRAMREAD

549

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

552

FRAMREAD ermöglicht das schnelle Lesen unterschiedlicher Datentypen direkt aus dem Anwender-Retain-Speicher (FRAM[1]).

Der FB liest den Inhalt ab der Adresse von SRC aus dem FRAM-Speicher. Dabei werden genau so viele Bytes übertragen, wie diese unter LEN angegeben sind.
Würde durch die angegebene Anzahl an Bytes der FRAM-Speicherbereich überschritten werden, werden nur die Daten bis zum Ende des FRAM-Speicherbereichs gelesen.

► Für die Zieladresse DST gilt:
 ! Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!

[1]) FRAM steht hier allgemein für alle Arten von nichtflüchtigen, schnellen Speichern.

Parameter der Eingänge

2606

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
SRC	DWORD	Relative Quell-Anfangsadresse im Speicher zulässig = 0... 16 383 = 0x0000 0000...0x0000 3FFF
LEN	DWORD	Anzahl der Datenbytes zulässig = 0...16 384 = 0x0000 0000...0x0000 4000
DST	DWORD	Anfangsadresse der Zielvariablen ! Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!

FRAMWRITE

543

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:

```
FRAMWRITE
ENABLE
DST
LEN
SRC
```

Beschreibung

546

FRAMWRITE ermöglicht das schnelle Schreiben unterschiedlicher Datentypen direkt in den Anwender-Retain-Speicher (FRAM¹).

Der FB schreibt den Inhalt ab der Adresse SRC in den spannungsausfallsicheren FRAM-Speicher. Dabei werden genau so viele Bytes übertragen, wie diese über LEN angegeben sind.
Würde durch die angegebene Anzahl an Bytes der FRAM-Speicherbereich überschritten werden, werden nur die Daten bis zum Ende des FRAM-Speicherbereichs geschrieben.

► Für die Quelladresse SRC gilt:
  [!] Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!

[!] Falls Zieladresse DST außerhalb des zulässigen Bereichs: kein Datentransfer!

¹) FRAM steht hier allgemein für alle Arten von nichtflüchtigen, schnellen Speichern.

Parameter der Eingänge

2605

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
DST	DWORD	Relative Zieladresse im Speicher zulässig = 0...16 383 = 0x0000 0000...0x0000 3FFF
LEN	DWORD	Anzahl der Datenbytes zulässig = 0...16 384 = 0x0000 0000...0x0000 4000
SRC	DWORD	Anfangsadresse der Quellvariablen [!] Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!

MEMCPY

409

= Memory Copy
Baustein-Typ = Funktionsbaustein (FB)
Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:

```
MEMCPY
DST
SRC
LEN
```

Beschreibung

15944
412

MEMCPY ermöglicht das Schreiben und Lesen unterschiedlicher Datentypen direkt in den Speicher.
Der FB schreibt den Inhalt ab der Adresse von SRC an die Adresse DST.

► Für die Adressen SRC und DST gilt:
  ! Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!

> Dabei werden genau so viele Bytes übertragen, wie diese unter LEN angegeben wurden. Dadurch ist es auch möglich, genau ein Byte einer Word-Variablen zu übertragen.

> Befindet sich der Speicherbereich, in den die Daten kopiert werden sollen, nicht komplett in einem zulässigen Speicherbereich, werden die Daten nicht kopiert und es wird ein Parameterfehler gemeldet.

DST Speicherbereich	Gerät	Speichergröße
Anwendungsdaten	(alle)	192 kBytes
Retain-Daten	CR0032, CR0232, CR0234. CR7n32	32 kBytes
	CR0033, CR0133, CR0233, CR0235	64 kBytes

Parameter der Eingänge

413

Parameter	Datentyp	Beschreibung
DST	DWORD	Startadresse im Zielspeicher ! Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!
SRC	DWORD	Startadresse im Quellspeicher ! Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!
LEN	WORD	Anzahl (≥ 1) der zu übertragenden Daten-Bytes

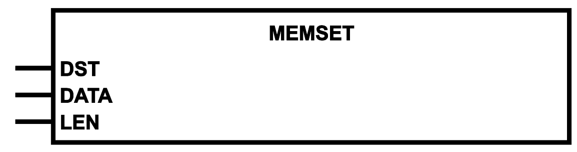
MEMSET

2348

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

2350

MEMSET ermöglicht das Beschreiben eines bestimmten Datenbereiches.

Der FB beschreibt den Speicher ab der Adresse DST mit der Anzahl von LEN Bytes mit dem Inhalt von DATA.

► Für die Ziel-Adresse DST gilt:
  Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!

> Befindet sich der Speicherbereich, in den die Daten kopiert werden sollen, nicht komplett in einem zulässigen Speicherbereich, werden die Daten nicht kopiert und es wird ein Parameterfehler gemeldet.

DST Speicherbereich	Gerät	Speichergröße
Anwendungsdaten	(alle)	192 kBytes
Retain-Daten	CR0032, CR0232, CR0234, CR7n32	32 kBytes
	CR0033, CR0133, CR0233, CR0235	64 kBytes

Parameter der Eingänge

2351

Parameter	Datentyp	Beschreibung
DST	DWORD	Startadresse im Zielspeicher Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!
DATA	BYTE	zu schreibender Wert
LEN	WORD	Anzahl der mit DATA zu beschreibenden Datenbytes

5.2.19 Bausteine: Datenzugriff und Datenprüfung

Inhalt

1598

Die Bausteine in diesem Kapitel steuern den Datenzugriff und ermöglichen ein Prüfen der Daten.

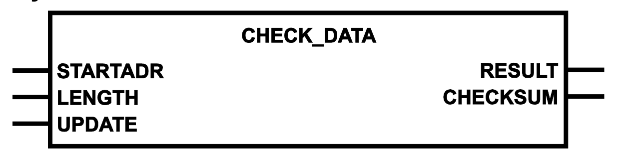
CHECK_DATA

603

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

606

CHECK_DATA erzeugt über einen konfigurierbaren Speicherbereich eine Prüfsumme (CRC) und prüft die Daten des Speicherbereichs auf ungewollte Veränderung.

- ► Für jeden zu überwachenden Speicherbereich eine eigene Instanz des FB erzeugen.
- ► (!) Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!
- ► Zusätzlich die Anzahl der Datenbytes LENGTH (Länge ab der STARTADR) angeben.

Ungewollte Änderung: Fehler!
Wenn Eingang UPDATE = FALSE und Daten im Speicher sich ungewollt verändern, wird RESULT = FALSE. Das Ergebnis kann dann für weitere Aktionen (z.B. Abschalten der Ausgänge) genutzt werden.

Gewollte Änderung:
Nur wenn der Eingang UPDATE auf TRUE gesetzt ist, sind Datenänderungen im Speicher (z.B. vom Anwendungsprogramm oder **ecomat*mobile***-Gerät) zulässig. Der Wert der Prüfsumme wird dann neu berechnet. Der Ausgang RESULT ist wieder permanent TRUE.

Parameter der Eingänge

2612

Parameter	Datentyp	Beschreibung
STARTADR	DWORD	Startadresse des überwachten Datenspeichers (WORD-Adresse ab %MW0) (!) Die Adresse mit dem Operator ADR ermitteln und dem FB übergeben!
LENGTH	DWORD	Länge des überwachten Datenspeichers in [Byte]
UPDATE	BOOL	TRUE: Daten wurden geändert > FB berechnet eine neue Prüfsumme FALSE: Daten wurden nicht geändert > FB prüft den Speicherbereich

Parameter der Ausgänge

2613

Parameter	Datentyp	Beschreibung
RESULT	BOOL	TRUE: CRC-Prüfsumme in Ordnung: Daten sind gewollt verändert oder nicht verändert FALSE: CRC-Prüfsumme fehlerhaft: Daten wurden ungewollt verändert
CHECKSUM	DWORD	aktuelle CRC-Prüfsumme

Beispiel: CHECK_DATA

4168

Im folgenden Beispiel ermittelt das Programm die Prüfsumme und legt sie über den Pointer pt im RAM ab:

```
0001 PROGRAM PLC_PRG
0002 VAR
0003     m1 : BOOL := TRUE;
0004     cd1 : CHECK_DATA;
0005     ok : BOOL;
0006     pt : POINTER TO WORD;
0007 END_VAR
0008
```

```
0001
16#82DC00 - ADD
16#400    -      - SUB - pt
             2   -

0002
                                     cd1
16#400 - SUB                     CHECK_DATA
    2  -     - 16#82DC00 - STARTADR      RESULT - ok
                       - LENGTH      CHECKSUM - pt
                    m1 - UPDATE

0003
FALSE —— m1
```

Das hier gezeigte Verfahren ist für den Flash-Speicher nicht geeignet.

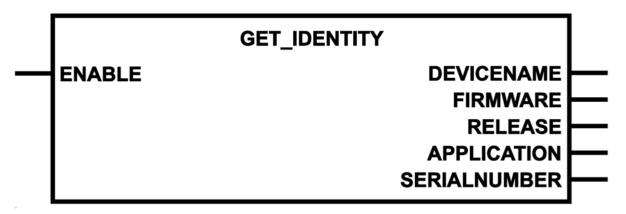
GET_IDENTITY

19287

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`
Neuer Ausgang SERIALNUMBER ist enthalten in:
- CR0032 ab LZS V02.01.06
- CR0033 ab LZS V01.00.09
- CR0133 ab LZS V01.00.09
- CR0232 ab LZS V01.00.03
- CR0233 ab LZS V01.00.09

Symbol in CODESYS:



Beschreibung

19288

GET_IDENTITY liest die im Gerät gespeicherten spezifischen Kennungen:
- Hardware-Name und Hardware-Version des Geräts
- Name des Laufzeitsystems im Gerät
- Version und Ausgabe des Laufzeitsystems im Gerät
- Name der Anwendung (wurde zuvor mit *SET_IDENTITY* (→ Seite 216) gespeichert)
- Seriennummer des Geräts

Parameter der Eingänge

2609

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert

Parameter der Ausgänge

19289

Parameter	Datentyp	Beschreibung
DEVICENAME	STRING(31)	Hardware-Name und Hardware-Version des Geräts als Zeichenkette von max. 31 Zeichen z.B.: "CR0403 01.00.00"
FIRMWARE	STRING(31)	Name des Laufzeitsystems im Gerät als Zeichenkette von max. 31 Zeichen z.B.: "CR0403"
RELEASE	STRING(31)	Version und Ausgabe des Laufzeitsystems im Gerät als Zeichenkette von max. 31 Zeichen z.B.: "V01.00.00 120215"
APPLICATION	STRING(79)	Name der Anwendung als String von max. 79 Zeichen z.B.: "Crane1704"
SERIALNUMBER	STRING(31)	Seriennummer des Geräts als Zeichenkette von max. 31 Zeichen z.B.: "12345678"

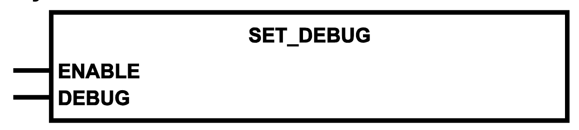
SET_DEBUG

290

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

293

SET_DEBUG organisiert den DEBUG-Modus ohne aktiven Test-Eingang (→ Kapitel *TEST-Betrieb* (→ Seite 48)).

Wird der Eingang DEBUG auf TRUE gesetzt, kann z.B. das Programmiersystem oder der Downloader mit dem Gerät kommunizieren und Systemkommandos ausführen (z.B. für Servicefunktionen über das GSM-Modem CANremote).

> ! Ein Software-Download ist in dieser Betriebsart nicht möglich, da der Test-Eingang nicht mit Versorgungsspannung verbunden wird. Nur lesender Zugriff ist möglich.

Parameter der Eingänge

294

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
DEBUG	BOOL	TRUE: Debugging über die Schnittstellen möglich FALSE: Debugging über die Schnittstellen nicht möglich

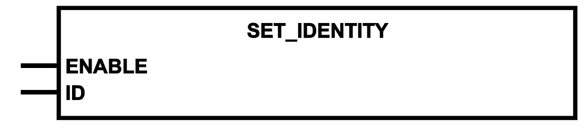
SET_IDENTITY

11927

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



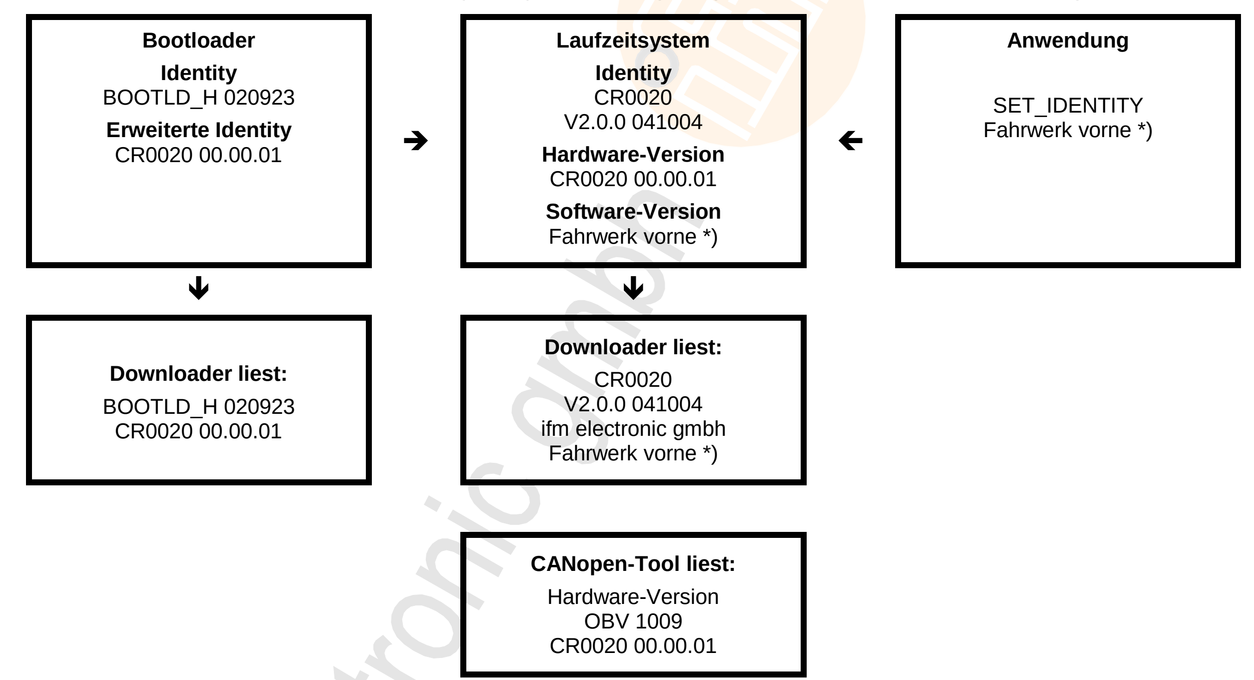
Beschreibung

287

SET_IDENTITY setzt eine anwendungsspezifische Programmkennung.

Mit dem FB kann durch das Anwendungsprogramm eine Programmkennung erzeugt werden. Diese Kennung kann zur Identifizierung des geladenen Programms über das Software-Tool DOWNLOADER.EXE als Software-Version ausgelesen werden.

Die nachfolgende Grafik zeigt die Zusammenhänge der unterschiedlichen Kennungen, wie sie mit den unterschiedlichen Software-Tools angezeigt werden. (Beispiel: ClassicController CR0020):



*) 'Fahrwerk vorne' steht hier stellvertretend für einen kundenspezifischen Text.

Parameter der Eingänge

11928

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
ID	STRING(79)	beliebiger Text mit einer maximalen Länge von 79 Zeichen

SET_PASSWORD

266

Baustein-Typ = Funktionsbaustein (FB)

Baustein ist enthalten in Bibliothek `ifm_CR0033_Vxxyyzz.LIB`

Symbol in CODESYS:



Beschreibung

269

SET_PASSWORD setzt Benutzerkennung für Programm- und Speicher-Upload mit dem DOWNLOADER.

Ist die Benutzerkennung aktiv, kann durch das Software-Tool DOWNLOADER das Anwendungsprogramm oder der Datenspeicher nur ausgelesen werden, wenn das richtige Password eingegeben wurde.

Wird an den Eingang PASSWORD ein Leer-String (Default-Zustand) übergeben, ist ein Upload des Anwendungsprogramms oder des Datenspeichers jederzeit möglich.

Ein neues Passwort wird nur nach dem Löschen des bisherigen Passwortes übernommen.

> ⓘ Beim Laden eines neuen Anwendungsprogramms wird die Kennung wieder zurückgesetzt.

Parameter der Eingänge

2353

Parameter	Datentyp	Beschreibung
ENABLE	BOOL	FALSE ⇨ TRUE (Flanke): Baustein initialisieren (nur 1 Zyklus) > Baustein-Eingänge lesen TRUE: Baustein ausführen FALSE: Baustein wird nicht ausgeführt > Baustein-Eingänge sind nicht aktiv > Baustein-Ausgänge sind nicht spezifiziert
PASSWORD	STRING(16)	Benutzerkennung Wenn PASSWORD = "", dann ist Zugriff ohne Passworteingabe möglich.

6 Diagnose und Fehlerbehandlung

Inhalt

19598

Das Laufzeitsystem (LZS) überprüft das Gerät durch interne Fehler-Checks:
• in der Startphase (Reset-Phase)
• während der Ausführung des Anwendungsprogramms
→ Kapitel *Betriebszustände* (→ Seite 44)

So wird eine möglichst hohe Betriebssicherheit gewährleistet.

6.1 Diagnose

19601

Bei der Diagnose wird der "Gesundheitszustand" des Gerätes geprüft. Es soll festgestellt werden, ob und gegebenenfalls welche →Fehler im Gerät vorhanden sind.

Je nach Gerät können auch die Ein- und Ausgänge auf einwandfreie Funktion überwacht werden:
- Drahtbruch,
- Kurzschluss,
- Wert außerhalb des Sollbereichs.

Zur Diagnose können Konfigurations-Dateien herangezogen werden, die während des "normalen" Betriebs des Gerätes erzeugt wurden.
Der korrekte Start der Systemkomponenten wird während der Initialisierungs- und Startphase überwacht.
Zur weiteren Diagnose können auch Selbsttests durchgeführt werden.

6.2 Fehler

19602

Ein Fehler ist die Unfähigkeit einer Einheit, eine geforderte Funktion auszuführen.
Kein Fehler ist diese Unfähigkeit während vorbeugender Wartung oder anderer geplanter Handlungen oder aufgrund des Fehlers externer Mittel.
Ein Fehler ist oft das Resultat eines Ausfalls der Einheit selbst, kann aber ohne vorherigen Ausfall bestehen.
In der ISO 13849-1 ist mit "Fehler" der "zufällige Fehler" gemeint.

6.3 Reaktion im Fehlerfall

19603
12217

Bei erkannten Fehlern kann im Anwendungsprogramm zusätzlich der Systemmerker ERROR gesetzt werden. Im Fehlerfall reagiert die Steuerung dann wie folgt:

> die Betriebs-LED leuchtet rot,

> die Ausgangsrelais schalten ab,

> die darüber gesicherten Ausgänge sind spannungsfrei,

> die logischen Signalzustände der Ausgänge ändern sich dadurch NICHT.

HINWEIS

Bei Abschalten der Ausgänge durch die Relais bleiben die logischen Signalzustände unverändert.

► Der Programmierer muss das ERROR-Bit auswerten und so im Fehlerfall die Ausgänge auch logisch zurücksetzen.

Vollständige Aufstellung der gerätespezifischen Fehler-Codes und Diagnosemeldungen
→ Kapitel *Systemmerker* (→ Seite 221)

6.4 Relais: wichtige Hinweise!

14034

ACHTUNG

Vorzeitiger Verschleiß der Relaiskontakte möglich.

► Im Normalfall die Relais nur lastfrei schalten!
Dazu via Anwendungsprogramm alle relevanten Ausgänge auf FALSE setzen!

6.5 Reaktion auf System-Fehler

14033

> ❗ Für die sichere Verarbeitung der Daten im Anwendungsprogramm ist allein dessen Programmierer verantwortlich.

► Die spezifischen Fehlermerker und / oder Fehler-Codes im Anwendungsprogramm verarbeiten!
Über den Fehlermerker/Fehler-Code erhält man eine Fehlerbeschreibung.
Dieser Fehlermerker/Fehler-Code kann bei Bedarf weiter verarbeitet werden.

Nach der Analyse und Beseitigung der Fehler-Ursache:

► Grundsätzlich alle Fehlermerker durch das Anwendungsprogramm zurücksetzen.
Ohne ausdrückliches Rücksetzen der Fehlermerker bleiben die Merker gesetzt mit entsprechender Auswirkung im Anwendungsprogramm.

6.6 CAN / CANopen: Fehler und Fehlerbehandlung

19604

→ Systemhandbuch "Know-How ecomat*mobile*"
→ Kapitel *CAN / CANopen: Fehler und Fehlerbehandlung*

7 Anhang

Inhalt

1664

Hier stellen wir Ihnen – ergänzend zu den Angaben in den Datenblättern – zusammenfassende Tabellen zur Verfügung.

7.1 Systemmerker

Inhalt

12167



Die zu den Systemmerkern gehörenden Merkeradressen können sich bei einer Erweiterung der Steuerungskonfiguration ändern.

► Für die Programmierung nur die Symbolnamen der Systemmerker nutzen!

→ Systemhandbuch "Know-How ecomat*mobile*"
→ Kapitel *Fehler-Codes und Diagnoseinformationen*

7.1.1 Systemmerker: CAN

12820

Systemmerker (Symbolname)	Typ	Beschreibung
CANx_BAUDRATE	WORD	CAN-Schnittstelle x: eingestellte Baudrate in [kBaud]
CANx_BUSOFF	BOOL	CAN-Schnittstelle x: Fehler "CAN-Bus off" Zurücksetzen des Fehler-Codes setzt auch den Merker zurück
CANx_DOWNLOADID	BYTE	CAN-Schnittstelle x: eingestellter Download-Identifier
CANx_ERRORCOUNTER_RX	BYTE	CAN-Schnittstelle x: Fehlerzähler Empfang Reset des Merkers ist via Schreibzugriff möglich
CANx_ERRORCOUNTER_TX	BYTE	CAN-Schnittstelle x: Fehlerzähler Versand Reset des Merkers ist via Schreibzugriff möglich
CANx_LASTERROR	BYTE	CAN-Schnittstelle x: Fehlernummer der letzten CAN-Übertragung: 0 = kein Fehler: Initial-Wert 1 = Stuff Error: mehr als 5 gleiche Bits in Reihe auf dem Bus 2 = Form Error: empfangenes Telegramm hatte falsches Format 3 = Ack Error: gesendetes Telegramm wurde nicht bestätigt 4 = Bit1 Error: außerhalb des Arbitrierungsbereichs wurde ein rezessives Bit gesendet, aber ein dominates Bit auf dem Bus gelesen 5 = Bit0 Error: es wurde versucht, ein dominantes Bit zu senden, aber es wurde ein rezessiver Pegel gelesen ODER: während Bus-off Recovery wurde eine Sequenz von 11 rezessiven Bits gelesen 6 = CRC Error: die Prüfsumme der empfangenen Nachricht war falsch
CANx_WARNING	BOOL	CAN-Schnittstelle x: Warnschwelle erreicht (≥ 96) Reset des Merkers ist via Schreibzugriff möglich

CANx steht für x = 1...4 = Nummer der CAN-Schnittstelle

7.1.2 Systemmerker: SAE-J1939

12815

Systemmerker (Symbolname)	Typ	Beschreibung
J1939_RECEIVE_OVERWRITE	BOOL	Einstellung gilt nur für J1939 Daten, die nicht über ein J1939-Transportprotokoll übertragen wurden. TRUE: Alte Daten werden durch die neuen Daten überschrieben, wenn die alten Daten noch nicht aus der Funktionsbaustein-Instanz ausgelesen wurden FALSE: Neue Daten werden verworfen, solange die alten Daten noch nicht aus der Funktionsbaustein-Instanz ausgelesen wurden ℹ Neue Daten können eintreffen, bevor die alten ausgelesen wurden, wenn der IEC-Zyklus länger ist als die Aktualisierungsfrequenz der J1939-Daten
J1939_TASK	BOOL	Mit J1939_TASK wird die Zeitanforderung beim Versenden von J1939-Telegrammen eingehalten. Sollen J1939-Telegramme mit einer Wiederholzeit ≤ 50 ms versendet werden, setzt das Laufzeitsystem automatisch J1939_TASK=TRUE. Für Anwendungen, bei denen die Zeitanforderungen ≥ SPS-Zykluszeit sind: ► Systemlast reduzieren mit J1939_TASK=FALSE! TRUE: J1939-Task ist aktiv (= Initialwert) Der Task wird alle 2 ms aufgerufen Der J1939-Stack sendet seine Telegramme im benötigten Zeitraster FALSE: J1939-Task ist nicht aktiv

7.1.3 Systemmerker: Fehlermerker (Standard-Seite)

2379

Systemmerker (Symbolname)	Typ	Beschreibung
ERROR	BOOL	TRUE: sicherer Zustand eingenommen alle Ausgänge = AUS Ausgangs-Relais = AUS (z.B. fataler Fehler / Error-Stop) FALSE: kein schwerer Fehler aufgetreten
ERROR_BREAK_Ix (x=0...n; Wert abhängig vom Gerät, → Datenblatt)	DWORD	Eingangsgruppe x: Leiterbruch-Fehler oder (Widerstandseingang): Schluss nach Versorgung [Bit 0 für Eingang 0] ... [Bit z für Eingang z] dieser Gruppe Bit = TRUE: Fehler Bit = FALSE: kein Fehler
ERROR_BREAK_Qx (x=0...n; Wert abhängig vom Gerät, → Datenblatt)	DWORD	Ausgangsgruppe x: Leiterbruch-Fehler [Bit 0 für Ausgang 0] ... [Bit z für Ausgang z] dieser Gruppe Bit = TRUE: Fehler Bit = FALSE: kein Fehler
ERROR_CONTROL_Qx (x=0...n; Wert abhängig vom Gerät, → Datenblatt)	DWORD	Ausgangsgruppe x: Fehler Stromregelung Endwert kann nicht erreicht werden [Bit 0 für Ausgang 0] ... [Bit z für Ausgang z] dieser Gruppe Bit = TRUE: Fehler Bit = FALSE: kein Fehler
ERROR_CURRENT_Ix (x=0...n; Wert abhängig vom Gerät, → Datenblatt)	DWORD	Eingangsgruppe x: Überstrom-Fehler [Bit 0 für Eingang 0] ... [Bit z für Eingang z] dieser Gruppe Bit = TRUE: Fehler Bit = FALSE: kein Fehler
ERROR_IO	BOOL	Sammelfehlermeldung Ein-/Ausgangsfehler TRUE: Fehler FALSE: kein Fehler
ERROR_POWER	BOOL	Überspannungs-Fehler für VBBS / Klemme 15: TRUE: Wert außerhalb des zulässigen Bereichs oder: Differenz (VBB15 - VBBS) zu groß > allgemeiner Fehler FALSE: Wert in Ordnung
ERROR_SHORT_Ix (x=0...n; Wert abhängig vom Gerät, → Datenblatt)	DWORD	Eingangsgruppe x: Kurzschluss-Fehler [Bit 0 für Eingang 0] ... [Bit z für Eingang z] dieser Gruppe Bit = TRUE: Fehler Bit = FALSE: kein Fehler
ERROR_SHORT_Qx (x=0...n; Wert abhängig vom Gerät, → Datenblatt)	DWORD	Ausgangsgruppe x: Kurzschluss-Fehler oder Überlast-Fehler [Bit 0 für Ausgang 0] ... [Bit z für Ausgang z] dieser Gruppe Bit = TRUE: Fehler Bit = FALSE: kein Fehler
ERROR_TEMPERATURE	BOOL	Temperatur-Fehler TRUE: Wert außerhalb des zulässigen Bereichs > allgemeiner Fehler FALSE: Wert in Ordnung
ERROR_VBBx	BOOL	Versorgungsspannungs-Fehler an VBBx (x = O | R): TRUE: Wert außerhalb des zulässigen Bereichs > allgemeiner Fehler FALSE: Wert in Ordnung
ERRORCODE	DWORD	Zuletzt eingetragener Fehler in der internen Fehlerliste Die Liste enthält alle aufgetretenen Fehler-Codes.
LAST_RESET	BYTE	Grund für den letzten Reset: 00 = Reset der Anwendung 01 = PowerOn-Reset 02 = Watchdog-Reset 03 = Soft-Reset 04 = Grund nicht feststellbar

7.1.4 Systemmerker: LED (Standard-Seite)

12817

Systemmerker (Symbolname)	Typ	Beschreibung
LED	WORD	LED-Farbe für "LED eingeschaltet": 0x0000 = LED_GREEN (voreingestellt) 0x0001 = LED_BLUE 0x0002 = LED_RED 0x0003 = LED_WHITE 0x0004 = LED_BLACK 0x0005 = LED_MAGENTA 0x0006 = LED_CYAN 0x0007 = LED_YELLOW
LED_X	WORD	LED-Farbe für "LED ausgeschaltet": 0x0000 = LED_GREEN 0x0001 = LED_BLUE 0x0002 = LED_RED 0x0003 = LED_WHITE 0x0004 = LED_BLACK (voreingestellt) 0x0005 = LED_MAGENTA 0x0006 = LED_CYAN 0x0007 = LED_YELLOW
LED_MODE	WORD	LED-Blinkfrequenz: 0x0000 = LED_2HZ (blinkt mit 2 Hz; voreingestellt) 0x0001 = LED_1HZ (blinkt mit 1 Hz) 0x0002 = LED_05HZ (blinkt mit 0,5 Hz) 0x0003 = LED_0HZ (leuchtet dauernd mit Wert in LED) 0x0004 = LED_5HZ (blinkt mit 5 Hz)

7.1.5 Systemmerker: Spannungen (Standard-Seite)

12135

Systemmerker (Symbolname)	Typ	Beschreibung
CLAMP_15_VOLTAGE	WORD	Spannung an Klemme 15 in [mV]
REF_VOLTAGE	WORD	Spannung am Referenzspannungsausgang in [mV]
REFERENCE_VOLTAGE_5	BOOL	Referenzspannungsausgang mit 5 V aktiviert
REFERENCE_VOLTAGE_10	BOOL	Referenzspannungsausgang mit 10 V aktiviert
RELAIS_VBBy y = O | R	BOOL	TRUE: Relais für VBBy aktiviert Ausgangsgruppe x wird mit Spannung versorgt (x = 1 | 2) FALSE: Relais für VBBy ausgeschaltet Ausgangsgruppe x ist spannungslos
SERIAL_MODE	BOOL	serielle Schnittstelle (RS232) für die Verwendung in der Anwendung aktivieren TRUE: RS232-Schnittstelle kann in der Anwendung verwendet werden, jedoch nicht mehr zum Programmieren, Debuggen oder Monitoren des Geräts. FALSE: RS232-Schnittstelle kann in der Anwendung nicht verwendet werden. Programmieren, Debuggen oder Monitoren des Geräts ist möglich.
SUPPLY_SWITCH	BOOL	Bit zum Abschalten der Versorgungs-Selbsthaltung VBBS. Das Rücksetzen des Merkers wird vom Laufzeitsystem nur akzeptiert, wenn die Spannung an Klemme 15 < 4 V ist, ansonsten wird der Merker wieder aktiviert. Die Trennung von VBBS erfolgt vor dem Beginn des nächsten SPS-Zyklus. Abhängig vom Ladezustand der internen Kondensatoren kann es noch eine gewisse Zeit dauern, bis das Gerät abschaltet. TRUE: Versorgung des Geräts über VBBS ist aktiv FALSE: Versorgung des Geräts über VBBS wird deaktiviert
SUPPLY_VOLTAGE	WORD	Versorgungsspannung an VBBS in [mV]
TEST	BOOL	TRUE: Test-Eingang ist aktiv: • Programmiermodus ist freigeben • Software-Download ist möglich • Zustand des Anwendungsprogramms ist abfragbar • kein Schutz der gespeicherten Software möglich FALSE: laufender Betrieb der Anwendung
VBBx_RELAIS_VOLTAGE x = O | R	WORD	Versorgungsspannung an VBBx nach Relaiskontakt in [mV]
VBBx_VOLTAGE x = O | R	WORD	Versorgungsspannung an VBBx in [mV]

7.1.6 Systemmerker: 16 Eingänge und 16 Ausgänge (Standard-Seite)

13121

Systemmerker (Symbolname)	Typ	Beschreibung
ANALOGxx xx = 00...15	WORD	Analog-Eingang xx: gefilterter A/D-Wandler-Rohwert (12 Bit) ohne Kalibrierung und Normierung
ANALOG_IRQxx xx = 00...07	WORD	Analogeingang Kanal xx: ungefilterter Rohwert der Spannung Verwendung im FB *SET_INTERRUPT_I* (→ Seite 124) oder *SET_INTERRUPT_XMS* (→ Seite 126)
CURRENTxx xx = 00...15	WORD	PWM-Ausgang xx: gefilterte A/D-Wandler-Rohwerte (12 Bit) der Strommessung ohne Kalibrierung und Normierung
Ixx xx = 00...15	BOOL	Status am Binäreingang xx Voraussetzung: Eingang ist als Binäreingang konfiguriert (MODE = IN_DIGITAL_H oder IN_DIGITAL_L) TRUE: Spannung am Binäreingang > 70 % von VBBS FALSE: Spannung am Binäreingang < 30 % von VBBS oder: nicht als Binäreingang konfiguriert oder: falsch konfiguriert
Ixx_DFILTER xx = 00...11	DWORD	Impulseingang xx: Impulsdauer in [µs], die als Glitch ignoriert werden soll. Die Erfassung des Eingangssignals verzögert sich um die eingestellte Zeit. zugelassen = 0...100 000 µs voreingestellt = 0 µs = kein Filter
Ixx_FILTER xx = 00...15	BYTE:=4	Binär- und Analogeingang xx: Grenzfrequenz (oder Signalanstiegszeit) des Software-Tiefpass-Filters erster Ordnung 0 = 0x00 = kein Filter 1 = 0x01 = 390 Hz (1 ms) 2 = 0x02 = 145 Hz (2,5 ms) 3 = 0x03 = 68 Hz (5 ms) 4 = 0x04 = 34 Hz (10 ms) (voreingestellt) 5 = 0x05 = 17 Hz (21 ms) 6 = 0x06 = 8 Hz (42 ms) 7 = 0x07 = 4 Hz (84 ms) 8 = 0x08 = 2 Hz (169 ms) größer = → voreingestellter Wert
Qxx xx = 00...15	BOOL	Status am Binärausgang xx: Voraussetzung: Ausgang ist als Binärausgang konfiguriert TRUE: Ausgang aktiviert FALSE: Ausgang deaktiviert (= Initialwert) oder: nicht als Binärausgang konfiguriert
Qxx_FILTER xx = 00...15	BYTE	Ausgang xx: Grenzfrequenz des Software-Tiefpass-Filters erster Ordnung für die Strommessung 0 = 0x00 = kein Filter 1 = 0x01 = 580 Hz (0,6 ms) 2 = 0x02 = 220 Hz (1,6 ms) 3 = 0x03 = 102 Hz (3,5 ms) 4 = 0x04 = 51 Hz (7 ms) (voreingestellt) 5 = 0x05 = 25 Hz (14 ms) 6 = 0x06 = 12 Hz (28 ms) 7 = 0x07 = 6 Hz (56 ms) 8 = 0x08 = 3 Hz (112 ms) größer = → voreingestellter Wert

7.2 Adressbelegung und E/A-Betriebsarten

Inhalt

1656

→ auch Datenblatt

7.2.1 Adressbelegung Ein-/Ausgänge

Inhalt

2371

Eingänge: Adressbelegung (Standard-Seite) (16 Eingänge)

10429

Abkürzungen → Kapitel *Hinweise zur Anschlussbelegung* (→ Seite 31)
Betriebsarten der Ein- und Ausgänge → Kapitel *Mögliche Betriebsarten Ein-/Ausgänge* (→ Seite 232)

IEC-Adresse	Symbolische Adresse
%IX0.0 %IW2	I00 ANALOG00
%IX0.1 %IW3	I01 ANALOG01
%IX0.2 %IW4	I02 ANALOG02
%IX0.3 %IW5	I03 ANALOG03
%IX0.4 %IW6	I04 ANALOG04
%IX0.5 %IW7	I05 ANALOG05
%IX0.6 %IW8	I06 ANALOG06
%IX0.7 %IW9	I07 ANALOG07
%IX0.8 %IW10	I08 ANALOG08
%IX0.9 %IW11	I09 ANALOG09
%IX0.10 %IW12	I10 ANALOG10
%IX0.11 %IW13	I11 ANALOG11
%IX0.12 %IW14	I12 ANALOG12
%IX0.13 %IW15	I13 ANALOG13
%IX0.14 %IW16	I14 ANALOG14
%IX0.15 %IW17	I15 ANALOG15

Ausgänge: Adressbelegung (Standard-Seite) (16 Ausgänge)

10430

Abkürzungen → Kapitel *Hinweise zur Anschlussbelegung* (→ Seite 31)
Betriebsarten der Ein- und Ausgänge → Kapitel *Mögliche Betriebsarten Ein-/Ausgänge* (→ Seite 232)

IEC-Adresse	Symbolische Adresse
%QX0.0 %IW18	Q00 CURRENT00
%QX0.1 %IW19	Q01 CURRENT01
%QX0.2 %IW20	Q02 CURRENT02
%QX0.3 %IW21	Q03 CURRENT03
%QX0.4 %IW22	Q04 CURRENT04
%QX0.5 %IW23	Q05 CURRENT05
%QX0.6 %IW24	Q06 CURRENT06
%QX0.7 %IW25	Q07 CURRENT07
%QX0.8 %IW26	Q08 CURRENT08
%QX0.9 %IW27	Q09 CURRENT09
%QX0.10 %IW28	Q10 CURRENT10
%QX0.11 %IW29	Q11 CURRENT11
%QX0.12 %IW30	Q12 CURRENT12
%QX0.13 %IW31	Q13 CURRENT13
%QX0.14 %IW32	Q14 CURRENT14
%QX0.15 %IW33	Q15 CURRENT15

7.2.2 Mögliche Betriebsarten Ein-/Ausgänge

Inhalt

2386

Eingänge: Betriebsarten (Standard-Seite) (16 Eingänge)

10431

Mögliche Konfigurations-Kombinationen (wo zulässig) entstehen durch Addition der Konfigurations-Werte.

= diese Konfiguration ist voreingestellt

Eingänge	mögliche Betriebsart		einstellen mit FB	FB-Eingang	Wert	
					dez	hex
I00...I11	IN_NOMODE	Aus	INPUT_ANALOG SET_INPUT_MODE	MODE	0	0000
	IN_DIGITAL_H	plus	INPUT_ANALOG SET_INPUT_MODE	MODE	1	0001
	IN_DIGITAL_L	minus	INPUT_ANALOG SET_INPUT_MODE	MODE	2	0002
	IN_CURRENT	0...20 000 µA	INPUT_ANALOG SET_INPUT_MODE	MODE	4	0004
	IN_VOLTAGE10	0...10 000 mV	INPUT_ANALOG SET_INPUT_MODE	MODE	8	0008
	IN_VOLTAGE30	0...30 000 mV	INPUT_ANALOG SET_INPUT_MODE	MODE	16	0010
	IN_RATIO	0...1 000 ‰	INPUT_ANALOG SET_INPUT_MODE	MODE	32	0020
	IN_DIAGNOSTIC	bei IN_DIGITAL_H	intern		64	0040
	IN_FAST	bei IN_DIGITAL_H	intern		128	0080
	Diagnose	bei IN_DIGITAL_H	SET_INPUT_MODE	DIAGNOSTICS	TRUE	
	IN_FAST	0...30 000 Hz	FREQUENCY FREQUENCY_PERIOD PHASE	Frequenzmessung		
		0,1...5 000 Hz	PERIOD	Periodendauermessung		
		0,1...5 000 Hz	PERIOD_RATIO	Periodendauer- und Ratiomessung		
		0...30 000 Hz	FAST_COUNT	Zähler		
		0...30 000 Hz	INC_ENCODER	Drehgeber erfassen		
I12...I14	IN_NOMODE	Aus	INPUT_ANALOG SET_INPUT_MODE	MODE	0	0000
	IN_DIGITAL_H	plus	INPUT_ANALOG SET_INPUT_MODE	MODE	1	0001
	IN_RESISTANCE	16...30 000 Ω	INPUT_ANALOG SET_INPUT_MODE	MODE	512	0200
I15	IN_NOMODE	Aus	INPUT_ANALOG SET_INPUT_MODE	MODE	0	0000
	IN_DIGITAL_H	plus	INPUT_ANALOG SET_INPUT_MODE	MODE	1	0001

Eingänge	mögliche Betriebsart		einstellen mit FB	FB-Eingang	Wert	
					dez	hex
	IN_RESISTANCE	3...680 Ω	INPUT_ANALOG SET_INPUT_MODE	MODE	512	0200

Betriebsarten mit folgendem Funktionsbaustein einstellen:

FAST_COUNT (→ Seite 140)	Zählerbaustein für schnelle Eingangsimpulse
FREQUENCY (→ Seite 142)	misst die Frequenz des am gewählten Kanal ankommenden Signals
FREQUENCY_PERIOD (→ Seite 144)	misst die Frequenz und die Periodendauer (Zykluszeit) in [µs] am angegebenen Kanal
INC_ENCODER (→ Seite 146)	Vorwärts-/Rückwärts-Zählerfunktion zur Auswertung von Drehgebern
INPUT_ANALOG (→ Seite 129)	Strom- und Spannungsmessung am analogen Eingangskanal
PERIOD (→ Seite 148)	misst am angegebenen Kanal die Frequenz und die Periodendauer (Zykluszeit) in [µs]
PERIOD_RATIO (→ Seite 150)	misst die Frequenz und die Periodendauer (Zykluszeit) in [µs] über die angegebenen Perioden am angegebenen Kanal. Zusätzlich wird das Puls-/Periodenverhältnis in [‰] angegeben.
PHASE (→ Seite 152)	liest ein Kanalpaar mit schnellen Eingängen ein und vergleicht die Phasenlage der Signale
SET_INPUT_MODE (→ Seite 132)	weist einem Eingangskanal eine Betriebsart zu

Ausgänge: Betriebsarten (Standard-Seite) (16 Ausgänge)

15523

= diese Konfiguration ist voreingestellt

Ausgänge	mögliche Betriebsart		einstellen mit FB	FB-Eingang	Wert	
					dez	hex
Q00...Q15	OUT_DIGITAL_H	plus	SET_OUTPUT_MODE	MODE	1	0001
	OUT_DIGITAL_L	minus	SET_OUTPUT_MODE	MODE	2	0002
	Diagnose	bei OUT_DIGITAL_H via Strommessung	SET_OUTPUT_MODE	DIAGNOSTICS	TRUE	
	Überlastschutz	bei OUT_DIGITAL_H mit Strommessung	SET_OUTPUT_MODE	PROTECTION	TRUE	
	Strommessbereich	keine Strommessung	SET_OUTPUT_MODE	CURRENT_RANGE	0	00
		2 A / 3 A	SET_OUTPUT_MODE	CURRENT_RANGE	1	01
		4 A	SET_OUTPUT_MODE	CURRENT_RANGE	2	02

Betriebsarten mit folgendem Funktionsbaustein einstellen:

OUTPUT_BRIDGE (→ Seite 159)	H-Brücke an einem PWM-Kanalpaar
OUTPUT_CURRENT_CONTROL (→ Seite 164)	Stromregler für einen PWMi-Ausgangskanal
PWM1000 (→ Seite 167)	initialisiert und parametriert einen PWM-fähigen Ausgangskanal das Puls-Pausen-Verhältnis kann in 1 ‰-Schritten angegeben werden
SET_OUTPUT_MODE (→ Seite 155)	setzt die Betriebsart des gewählten Ausgangskanals

Ausgänge: zulässige Betriebsarten

15525

Betriebsart		Q00	Q01	Q02	Q03	Q04	Q05	Q06	Q07
OUT_NOMODE	Aus	X	X	X	X	X	X	X	X
OUT_DIGITAL_H	plus	X	X	X	X	X	X	X	X
OUT_DIGITAL_L	minus	--	X	--	X	--	--	--	--
Diagnose	bei OUT_DIGITAL_H via Strommessung	X	X	X	X	X	X	X	X
Überlastschutz	bei OUT_DIGITAL_H mit Strommessung	X	X	X	X	X	X	X	X
Strommessbereich	2 A	X	X	X	X	--	--	--	--
	3 A	--	--	--	--	X	X	X	X
	4 A	X	X	X	X	--	--	--	--
PWM		X	X	X	X	X	X	X	X
PWMi		X	X	X	X	X	X	X	X
H-Brücke		--	X	--	X	--	--	--	--

Betriebsart		Q08	Q09	Q10	Q11	Q12	Q13	Q14	Q15
OUT_NOMODE	Aus	X	X	X	X	X	X	X	X
OUT_DIGITAL_H	plus	X	X	X	X	X	X	X	X
OUT_DIGITAL_L	minus	--	X	--	X	--	--	--	--
Diagnose	bei OUT_DIGITAL_H via Strommessung	X	X	X	X	X	X	X	X
Überlastschutz	bei OUT_DIGITAL_H mit Strommessung	X	X	X	X	X	X	X	X
Strommessbereich	2 A	X	X	X	X	--	--	--	--
	3 A	--	--	--	--	X	X	X	X
	4 A	X	X	X	X	--	--	--	--
PWM		X	X	X	X	X	X	X	X
PWMi		X	X	X	X	X	X	X	X
H-Brücke		--	X	--	X	--	--	--	--

7.2.3 Adressen / Variablen der E/As

Inhalt

2376

Eingänge: Adressen und Variablen (Standard-Seite) (16 Eingänge)

10432

IEC-Adresse	E/A-Variable	Bemerkung
%IB0		Eingangsbyte 0 (%IX0.0...%IX0.7)
%IB1		Eingangsbyte 1 (%IX0.8...%IX0.15)
%IW0		Eingangswort 0 (%IX0.0...%IX0.15)
%IW2	ANALOG00	Analogeingang Kanal 0
%IW3	ANALOG01	Analogeingang Kanal 1
%IW4	ANALOG02	Analogeingang Kanal 2
%IW5	ANALOG03	Analogeingang Kanal 3
%IW6	ANALOG04	Analogeingang Kanal 4
%IW7	ANALOG05	Analogeingang Kanal 5
%IW8	ANALOG06	Analogeingang Kanal 6
%IW9	ANALOG07	Analogeingang Kanal 7
%IW10	ANALOG08	Analogeingang Kanal 8
%IW11	ANALOG09	Analogeingang Kanal 9
%IW12	ANALOG10	Analogeingang Kanal 10
%IW13	ANALOG11	Analogeingang Kanal 11
%IW14	ANALOG12	Analogeingang Kanal 12
%IW15	ANALOG13	Analogeingang Kanal 13
%IW16	ANALOG14	Analogeingang Kanal 14
%IW17	ANALOG15	Analogeingang Kanal 15
%IW18	CURRENT00	Ausgangsstrom (Rohwert) an Q00
%IW19	CURRENT01	Ausgangsstrom (Rohwert) an Q01
%IW20	CURRENT02	Ausgangsstrom (Rohwert) an Q02
%IW21	CURRENT03	Ausgangsstrom (Rohwert) an Q03
%IW22	CURRENT04	Ausgangsstrom (Rohwert) an Q04
%IW23	CURRENT05	Ausgangsstrom (Rohwert) an Q05
%IW24	CURRENT06	Ausgangsstrom (Rohwert) an Q06
%IW25	CURRENT07	Ausgangsstrom (Rohwert) an Q07
%IW26	CURRENT08	Ausgangsstrom (Rohwert) an Q08
%IW27	CURRENT09	Ausgangsstrom (Rohwert) an Q09
%IW28	CURRENT10	Ausgangsstrom (Rohwert) an Q10
%IW29	CURRENT11	Ausgangsstrom (Rohwert) an Q11
%IW30	CURRENT12	Ausgangsstrom (Rohwert) an Q12
%IW31	CURRENT13	Ausgangsstrom (Rohwert) an Q13

IEC-Adresse	E/A-Variable	Bemerkung
%IW32	CURRENT14	Ausgangsstrom (Rohwert) an Q14
%IW33	CURRENT15	Ausgangsstrom (Rohwert) an Q15
%IW34	SUPPLY_VOLTAGE	Versorgungsspannung an VBBS in [mV]
%IW35	CLAMP_15_VOLTAGE	Spannung Klemme 15
%IW36	VBBO_VOLTAGE	Versorgungsspannung an VBBO in [mV]
%IW37	VBBR_VOLTAGE	Versorgungsspannung an VBBR in [mV]
%IW38	VBBO_RELAIS_VOLTAGE	Versorgungsspannung VBBO nach Relaiskontakt in [mV]
%IW39	VBBR_RELAIS_VOLTAGE	Versorgungsspannung VBBR nach Relaiskontakt in [mV]
%IW40	REF_VOLTAGE	Spannung am Referenzausgang Pin 51
%IW41	ANALOG_IRQ00	Interrupt zu Analogeingang Kanal 0
%IW42	ANALOG_IRQ01	Interrupt zu Analogeingang Kanal 1
%IW43	ANALOG_IRQ02	Interrupt zu Analogeingang Kanal 2
%IW44	ANALOG_IRQ03	Interrupt zu Analogeingang Kanal 3
%IW45	ANALOG_IRQ04	Interrupt zu Analogeingang Kanal 4
%IW46	ANALOG_IRQ05	Interrupt zu Analogeingang Kanal 5
%IW47	ANALOG_IRQ06	Interrupt zu Analogeingang Kanal 6
%IW48	ANALOG_IRQ07	Interrupt zu Analogeingang Kanal 7
%MB7960	ERROR_CURRENT_I0	Fehler DWORD Überstrom
%MB7964	ERROR_SHORT_I0	Fehler DWORD Kurzschluss
%MB7968	ERROR_BREAK_I0	Fehler DWORD Leiterbruch

Ausgänge: Adressen und Variablen (Standard-Seite) (16 Ausgänge)

10433

IEC-Adresse	E/A-Variable	Bemerkung
%QW0		Ausgangswort 0 (%QX0.0...%QX0.15)
%QB0		Ausgangsbyte 0 (%QX0.0...%QX0.7)
%QB1		Ausgangsbyte 1 (%QX0.8...%QX0.15)
%QB2	REFERENCE_VOLTAGE_5	Aktivieren des Referenzspannungsausgangs mit 5 V
%QB3	REFERENCE_VOLTAGE_10	Aktivieren des Referenzspannungsausgangs mit 10 V
%QB68	I00_FILTER	Filterbyte für %IX0.0 / %IW2
%QB69	I01_FILTER	Filterbyte für %IX0.1 / %IW3
%QB70	I02_FILTER	Filterbyte für %IX0.2 / %IW4
%QB71	I03_FILTER	Filterbyte für %IX0.3 / %IW5
%QB72	I04_FILTER	Filterbyte für %IX0.4 / %IW6
%QB73	I05_FILTER	Filterbyte für %IX0.5 / %IW7
%QB74	I06_FILTER	Filterbyte für %IX0.6 / %IW8
%QB75	I07_FILTER	Filterbyte für %IX0.7 / %IW9
%QB76	I08_FILTER	Filterbyte für %IX0.8 / %IW2
%QB77	I09_FILTER	Filterbyte für %IX0.9 / %IW3
%QB78	I10_FILTER	Filterbyte für %IX0.10 / %IW4
%QB79	I11_FILTER	Filterbyte für %IX0.11 / %IW5
%QB80	I12_FILTER	Filterbyte für %IX0.12 / %IW6
%QB81	I13_FILTER	Filterbyte für %IX0.13 / %IW7
%QB82	I14_FILTER	Filterbyte für %IX0.14 / %IW8
%QB83	I15_FILTER	Filterbyte für %IX0.15 / %IW9
%QB84	Q00_FILTER	Filter-Byte für %QX0.0
%QB85	Q01_FILTER	Filter-Byte für %QX0.1
%QB86	Q02_FILTER	Filter-Byte für %QX0.2
%QB87	Q03_FILTER	Filter-Byte für %QX0.3
%QB88	Q04_FILTER	Filter-Byte für %QX0.4
%QB89	Q05_FILTER	Filter-Byte für %QX0.5
%QB90	Q06_FILTER	Filter-Byte für %QX0.6
%QB91	Q07_FILTER	Filter-Byte für %QX0.7
%QB92	Q08_FILTER	Filter-Byte für %QX0.8
%QB93	Q09_FILTER	Filter-Byte für %QX0.9
%QB94	Q10_FILTER	Filter-Byte für %QX0.10
%QB95	Q11_FILTER	Filter-Byte für %QX0.11
%QB96	Q12_FILTER	Filter-Byte für %QX0.12
%QB97	Q13_FILTER	Filter-Byte für %QX0.13
%QB98	Q14_FILTER	Filter-Byte für %QX0.14
%QB99	Q15_FILTER	Filter-Byte für %QX0.15
%QD25	I00_DFILTER	Filterwert Zähl-/Impulseingang 0

IEC-Adresse	E/A-Variable	Bemerkung
%QD26	I01_DFILTER	Filterwert Zähl-/Impulseingang 1
%QD27	I02_DFILTER	Filterwert Zähl-/Impulseingang 2
%QD28	I03_DFILTER	Filterwert Zähl-/Impulseingang 3
%QD29	I04_DFILTER	Filterwert Zähl-/Impulseingang 4
%QD30	I05_DFILTER	Filterwert Zähl-/Impulseingang 5
%QD31	I06_DFILTER	Filterwert Zähl-/Impulseingang 6
%QD32	I07_DFILTER	Filterwert Zähl-/Impulseingang 7
%QD33	I08_DFILTER	Filterwert Zähl-/Impulseingang 8
%QD34	I09_DFILTER	Filterwert Zähl-/Impulseingang 9
%QD35	I10_DFILTER	Filterwert Zähl-/Impulseingang 10
%QD36	I11_DFILTER	Filterwert Zähl-/Impulseingang 11
%MB7948	ERROR_SHORT_Q0	Fehler DWORD Kurzschluss
%MB7952	ERROR_BREAK_Q0	Fehler DWORD Leiterbruch
%MB7956	ERROR_CONTROL_Q0	Fehler DWORD Stromregelung

7.3 Integriertes E/A-Modul: Beschreibung

Inhalt

16418

7.3.1 Systembeschreibung E/A-Modul ExB01

Inhalt

16422

Hardware-Beschreibung E/A-Modul

Inhalt

16423

Hardware-Aufbau E/A-Modul

16425

EXTENDED
Versorgung
VBBS
VBB1
VBB2
Wert: PDO10 VBBS
Wert: PDO10 VBB1
Wert: PDO10 VBB2
OUT00...OUT07
OUT08...OUT15

Grafik: Prinzipaufbau der Versorgung

Status-LED E/A-Modul

16414

Die Betriebszustände werden durch die integrierte Status-LED (Voreinstellung) angezeigt.

LED-Farbe	Anzeige	Beschreibung
Gelb	kurzzeitig ein	Zustand = INIT
	Δt = 200 ms	
Grün	konstant ein	Zustand = PRE-OPERATIONAL
Grün	blinkt 2 Hz	Zustand = OPERATIONAL
	Δt = 200 ms	
Grün	blinkt 1 Puls	Zustand = STOP
	Δt = 200 ms	
Rot	konstant ein	Fehler: CAN busoff
Rot	blinkt 1 Puls	EMCY: CAN error warning
	Δt = 200 ms	
Rot	blinkt 2 Pulse	EMCY: guarding / heartbeat
	Δt = 200 ms	
Rot	blinkt 3 Pulse	EMCY: synch error
	Δt = 200 ms	

Eingänge des integrierten E/A-Moduls ExB01

Inhalt

16229

Analog-Eingänge

15444

Die Analog-Eingänge können über das Anwendungsprogramm konfiguriert werden. Der Messbereich kann zwischen folgenden Bereichen umgeschaltet werden:

- Stromeingang 0...20 mA
- Spannungseingang 0...10 V
- Spannungseingang 0...32 V
- Widerstandsmessung 16...30 000 Ω (Messung gegen GND)

Die Spannungsmessung kann auch ratiometrisch erfolgen (0...1000 ‰, über FBs einstellbar). Das bedeutet, ohne zusätzliche Referenzspannung können Potentiometer oder Joysticks ausgewertet werden. Ein Schwanken der Versorgungsspannung hat auf diesen Messwert keinen Einfluss.

Alternativ kann ein Analog-Kanal auch binär ausgewertet werden.

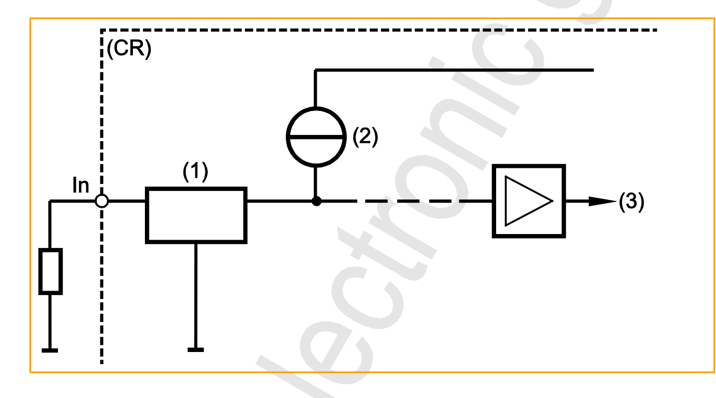
> ! Bei ratiometrischer Messung müssen die angeschlossenen Sensoren mit VBBS des Geräts versorgt werden. Dadurch werden Fehlmessungen durch Spannungsverschiebungen vermieden.

8971

Grafik: Prinzipschaltung Multifunktions-Eingang

In = Anschluss Multifunktions-Eingang n
(CR) = Gerät
(1) = Eingangsfilter
(2) = analoge Strommessung
(3a) = Binär-Eingang plus-schaltend
(3b) = Binär-Eingang minus-schaltend
(4a) = analoge Spannungsmessung 0...10 V
(4b) = analoge Spannungsmessung 0...32 V
(5) = Spannung
(6) = Referenz-Spannung

8972



Grafik: Prinzipschaltung Widerstandsmess-Eingang

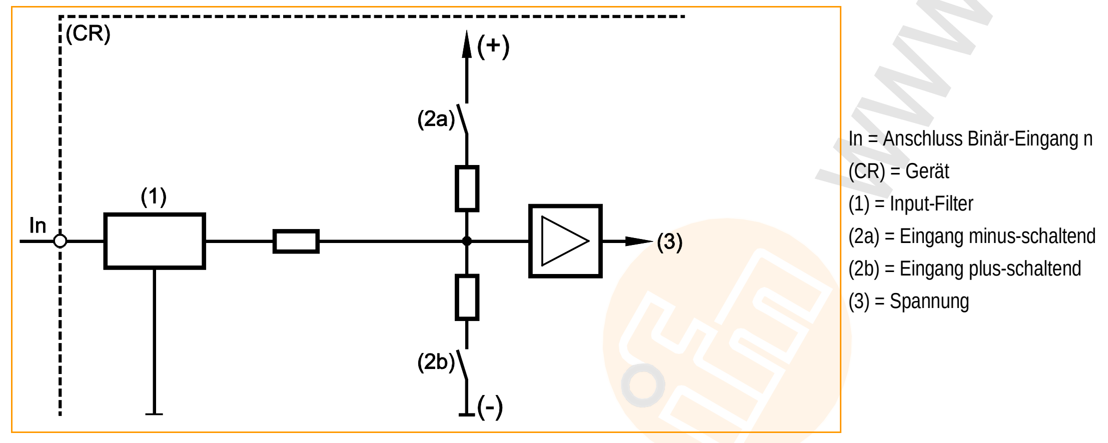
In = Anschluss Widerstandsmess-Eingang n
(CR) = Gerät
(1) = Eingangsfilter
(2) = Konstantstromquelle
(3) = Spannung

Binär-Eingänge

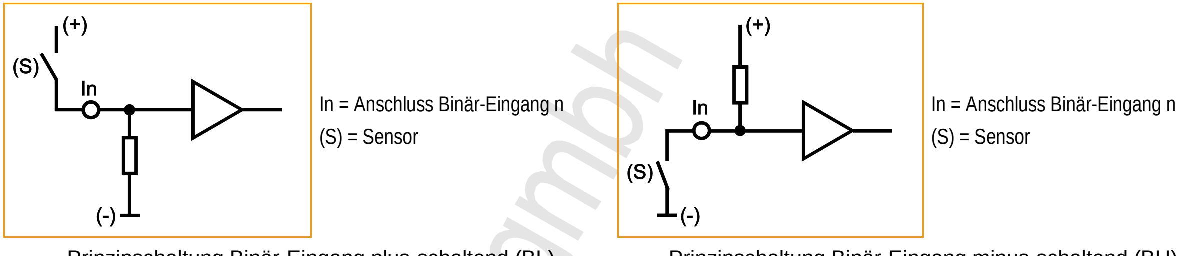
1015
7345

Der Binär-Eingang kann in folgenden Modi betrieben werden:
- binärer Eingang plus-schaltend (BL) für positives Gebersignal
- binärer Eingang, minus-schaltend (BH) für negatives Gebersignal

Je nach Gerät können auch die Binär-Eingänge unterschiedlich konfiguriert werden. Neben den Schutzmechanismen gegen Störungen werden die Binär-Eingänge intern über eine Analogstufe ausgewertet. Das ermöglicht die Diagnose der Eingangssignale. Im Anwendungsprogramm steht das Schaltsignal aber direkt als Bit-Information zur Verfügung.



Grafik: Prinzipschaltung Binär-Eingang minus-schaltend / plus-schaltend für negative und positive Gebersignale



Prinzipschaltung Binär-Eingang plus-schaltend (BL) für positives Sensorsignal:
Eingang = offen ⇨ Signal = Low (Supply)

Prinzipschaltung Binär-Eingang minus-schaltend (BH) für negatives Sensorsignal:
Eingang = offen ⇨ Signal = High (GND)

Bei einem Teil dieser Eingänge (→ Datenblatt) kann das Potential gewählt werden, gegen das geschaltet wird.

E/A-Modul Eingangsgruppe I0 = IN00...IN03

15801

Bei diesen Eingängen handelt es sich um eine Gruppe von Multifunktionskanälen.

Jeder einzelne dieser Eingänge ist wahlweise wie folgt konfigurierbar:
- analoger Eingang 0...20 mA
- analoger Eingang 0...10 V
- analoger Eingang 0...32 V
- binärer Eingang plus-schaltend (BL) für positives Gebersignal (mit/ohne Diagnose)
- binärer Eingang, minus-schaltend (BH) für negatives Gebersignal

→ Kapitel *Mögliche Betriebsarten E/A-Modul* (→ Seite 262)

Alle Eingänge zeigen das gleiche Verhalten bei Funktion und Diagnose.

► Die Konfiguration jedes einzelnen Eingangs erfolgt über die Steuerungskonfiguration:
→ Kapitel *Eingänge des integrierten E/A-Moduls konfigurieren* (→ Seite 257)

> Werden die Analogeingänge auf Strommessung konfiguriert, wird bei Überschreiten des Endwertes (23 mA für ≥ 40 ms) in den sicheren Spannungsmessbereich (0...32 V DC) geschaltet. Dies meldet das PDO1 als "overcurrent". Nach etwa einer Sekunde schaltet der Eingang selbsttätig auf den Strommessbereich zurück.

E/A-Modul Eingangsgruppe I1 = IN04...IN05

15803

Bei diesen Eingängen handelt es sich um eine Gruppe von Multifunktionskanälen.

Jeder einzelne dieser Eingänge ist wahlweise wie folgt konfigurierbar:
- binärer Eingang plus-schaltend (BL) für positives Gebersignal (mit/ohne Diagnose)
- Eingang für Widerstandsmessung (z.B. Temperatursensoren oder Tankgeber)

→ Kapitel *Mögliche Betriebsarten E/A-Modul* (→ Seite 262)

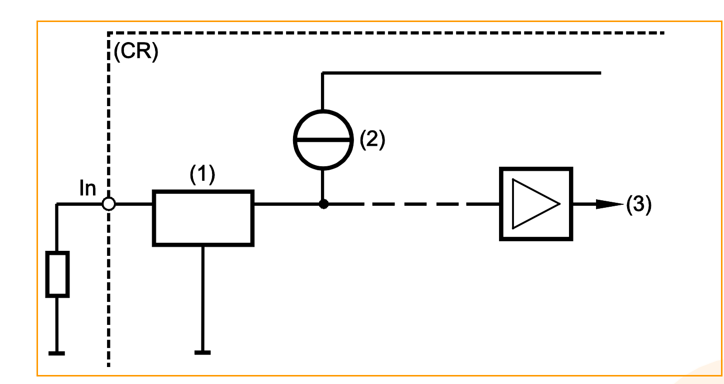
► Die Konfiguration jedes einzelnen Eingangs erfolgt über die Steuerungskonfiguration:
→ Kapitel *Eingänge des integrierten E/A-Moduls konfigurieren* (→ Seite 257)

Widerstandsmessung

9773

Typische Sensoren an diesen Eingängen:
- Tankpegel
- Temperatur (PT1000, NTC)

8972

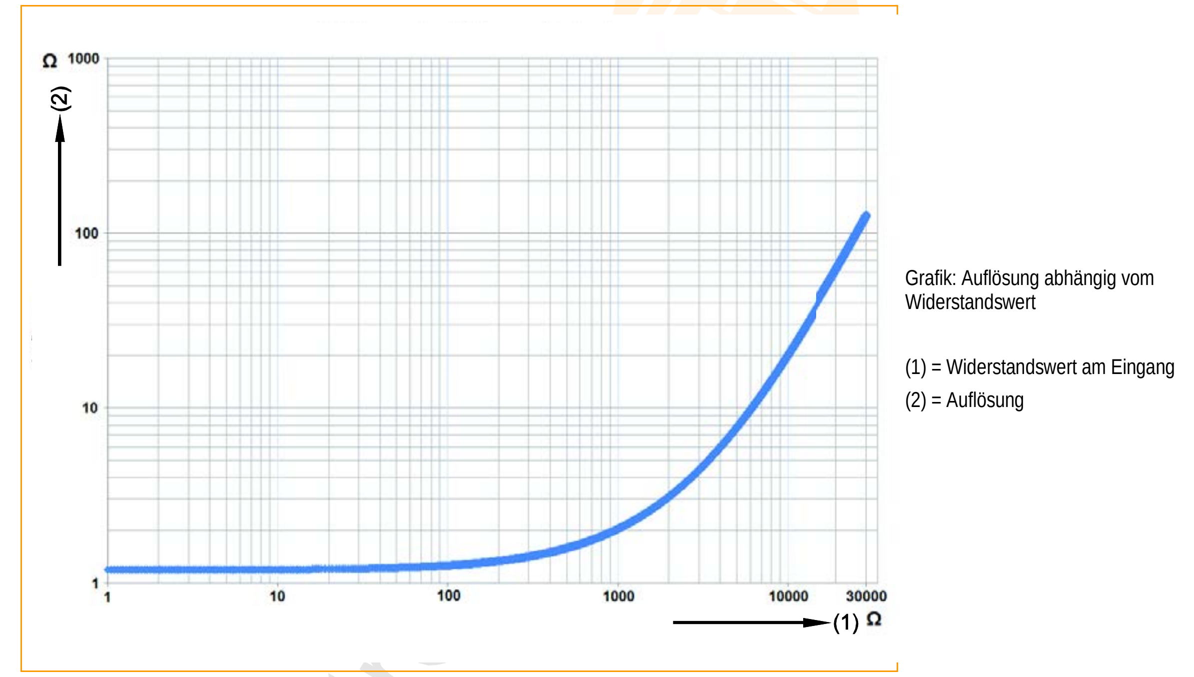


Grafik: Prinzipschaltung Widerstandsmess-Eingang

In = Anschluss Widerstandsmess-Eingang n
(CR) = Gerät
(1) = Eingangsfilter
(2) = Konstantstromquelle
(3) = Spannung

8970

Bei diesem Gerät ist die Auflösung nicht linear abhängig vom Widerstandswert, → Grafik:



Grafik: Auflösung abhängig vom Widerstandswert

(1) = Widerstandswert am Eingang
(2) = Auflösung

Um wieviel Ohm ändert sich der Messwert, wenn sich das Signal des A/D-Wandlers am Eingang um 1 ändert? Beispiele:
- Im Bereich 1...100 Ω beträgt die Auflösung 1,2 Ω.
- Im Bereich bei 1 kΩ beträgt die Auflösung ca. 2 Ω.
- Im Bereich bei 2 kΩ beträgt die Auflösung ca. 3 Ω.
- Im Bereich bei 3 kΩ beträgt die Auflösung ca. 6 Ω.
- Im Bereich bei 6 kΩ beträgt die Auflösung ca. 10 Ω.
- Im Bereich bei 10 kΩ beträgt die Auflösung ca. 11 Ω
- Im Bereich bei 20 kΩ beträgt die Auflösung ca. 60 Ω.

E/A-Modul Eingangsgruppe I2 = IN06...IN11

15804

Bei diesen Eingängen handelt es sich um eine Gruppe von Multifunktionskanälen.

Jeder einzelne dieser Eingänge ist wahlweise wie folgt konfigurierbar:
- binärer Eingang plus-schaltend (BL) für positives Gebersignal (mit/ohne Diagnose)

→ Kapitel *Mögliche Betriebsarten E/A-Modul* (→ Seite 262)

Diagnosefähige Sensoren nach NAMUR können ausgewertet werden.

► Die Konfiguration jedes einzelnen Eingangs erfolgt über die Steuerungskonfiguration:
→ Kapitel *Eingänge des integrierten E/A-Moduls konfigurieren* (→ Seite 257)

E/A-Modul Eingangsgruppe I3 = IN12...IN15

15805

Bei diesen Eingängen handelt es sich um eine Gruppe von Multifunktionskanälen.

Jeder einzelne dieser Eingänge ist wahlweise wie folgt konfigurierbar:
- binärer Eingang plus-schaltend (BL) für positives Gebersignal
- schneller Eingang für z.B. Inkrementalgeber und Frequenz- oder Periodendauermessung

→ Kapitel *Mögliche Betriebsarten E/A-Modul* (→ Seite 262)

► Die Konfiguration jedes einzelnen Eingangs erfolgt über die Steuerungskonfiguration:
→ Kapitel *Eingänge des integrierten E/A-Moduls konfigurieren* (→ Seite 257)

Ausgänge des integrierten E/A-Moduls ExB01

Inhalt

16234

E/A-Modul Ausgangsgruppe Q0 (OUT0, OUT1)

15806

Bei diesen Ausgängen handelt es sich um eine Gruppe von Multifunktionskanälen.

Jeder einzelne dieser Ausgänge ist wahlweise wie folgt konfigurierbar:
- binärer Ausgang, plus-schaltend (BH) mit Diagnosefunktion und Protection
- analoger Ausgang, stromgeregelt (PWMi)
- analoger Ausgang mit Pulsweitenmodulation (PWM)

→ Kapitel *Mögliche Betriebsarten E/A-Modul* (→ Seite 262)

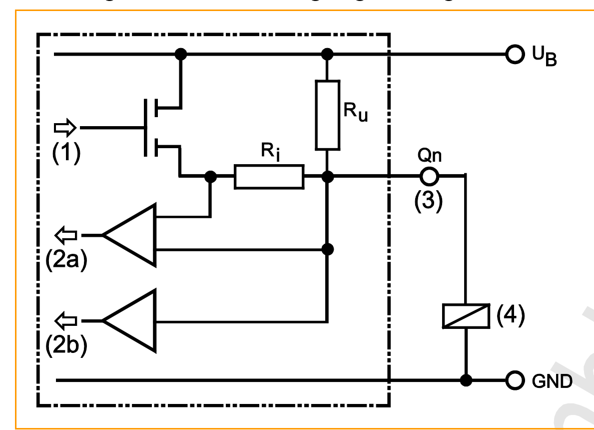
► Die Konfiguration jedes einzelnen Ausgangs erfolgt über die Steuerungskonfiguration:
→ Kapitel *Ausgänge des integrierten E/A-Moduls konfigurieren* (→ Seite 259)

► (!) Zu den Grenzwerten unbedingt das Datenblatt beachten!

Diagnose: binäre Ausgänge (via Strom- und Spannungsmessung)

19433
19434

Die Diagnose dieser Ausgänge erfolgt über eine interne Strom- und Spannungsmessung im Ausgang:



Grafik: Prinzipschaltung
(1) Ausgangskanal
(2a) Rücklesekanal für Diagnose via Strommessung
(2b) Rücklesekanal für Diagnose via Spannungsmessung
(3) Anschluss Ausgang
(4) Last

Diagnose: Überlast (via Strommessung)

19437
15249

Überlast kann nur an einem Ausgang mit Strommessung erkannt werden.

Überlast ist definiert als ...
"nominaler Maximalstrom laut Datenblatt + 12,5 %".

Diagnose: Leiterbruch (via Spannungsmessung)

19436
19404

Eine Leiterbruch-Erkennung erfolgt über den Rücklesekanal. Bei gesperrtem Ausgang (Qn=FALSE) wird dann ein Leiterbruch erkannt, wenn der Widerstand R_u den Rücklesekanal auf HIGH-Potential (VBB) zieht. Ohne den Leiterbruch würde die niederohmige Last (R_L < 10 kOhm) LOW (logisch 0) erzwingen.

Diagnose: Kurzschluss (via Spannungsmessung)

19405

Eine Kurzschluss-Erkennung erfolgt über den Rücklesekanal. Bei geschaltetem Ausgang (Qn=TRUE) wird dann ein Kurzschluss gegen GND erkannt, wenn der Rücklesekanal auf LOW-Potential (GND) gezogen wird.

E/A-Modul Ausgangsgruppe Q1 (OUT02...OUT07)

15808

Bei diesen Ausgängen handelt es sich um eine Gruppe von Multifunktionskanälen.

Jeder einzelne dieser Ausgänge ist wahlweise wie folgt konfigurierbar:
- binärer Ausgang, plus-schaltend (BH) mit/ohne Diagnosefunktion
- analoger Ausgang mit Pulsweitenmodulation (PWM)

→ Kapitel *Mögliche Betriebsarten E/A-Modul* (→ Seite 262)

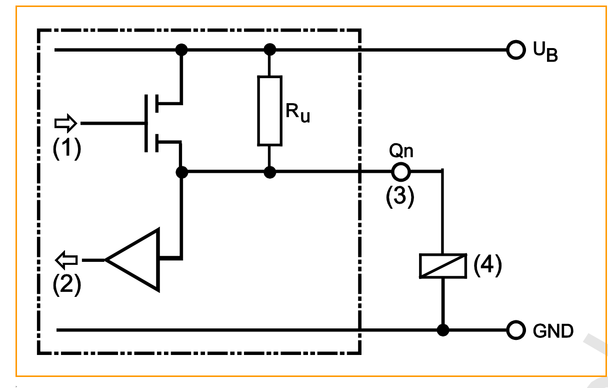
► Die Konfiguration jedes einzelnen Ausgangs erfolgt über die Steuerungskonfiguration:
→ Kapitel *Ausgänge des integrierten E/A-Moduls konfigurieren* (→ Seite 259)

► Zu den Grenzwerten unbedingt das Datenblatt beachten!

Diagnose: binäre Ausgänge (via Spannungsmessung)

19403
19397

Die Diagnose dieser Ausgänge erfolgt über eine interne Spannungsmessung im Ausgang:



Grafik: Prinzipschaltung
(1) Ausgangskanal
(2) Rücklesekanal für Diagnose
(3) Anschluss Ausgang n
(4) Last

Diagnose: Überlast

19448

Die Ausgänge haben keine Strommessung, keine Überlasterkennung.

Diagnose: Leiterbruch (via Spannungsmessung)

19404

Eine Leiterbruch-Erkennung erfolgt über den Rücklesekanal. Bei gesperrtem Ausgang (Qn=FALSE) wird dann ein Leiterbruch erkannt, wenn der Widerstand R_u den Rücklesekanal auf HIGH-Potential (VBB) zieht. Ohne den Leiterbruch würde die niederohmige Last (R_L < 10 kOhm) LOW (logisch 0) erzwingen.

Diagnose: Kurzschluss (via Spannungsmessung)

19405

Eine Kurzschluss-Erkennung erfolgt über den Rücklesekanal. Bei geschaltetem Ausgang (Qn=TRUE) wird dann ein Kurzschluss gegen GND erkannt, wenn der Rücklesekanal auf LOW-Potential (GND) gezogen wird.

E/A-Modul Ausgangsgruppe Q2 (OUT08...OUT09)

15809

Bei diesen Ausgängen handelt es sich um eine Gruppe von Multifunktionskanälen.

Jeder einzelne dieser Ausgänge ist wahlweise wie folgt konfigurierbar:
- binärer Ausgang, plus-schaltend (BH)
- analoger Ausgang mit Pulsweitenmodulation (PWM)
- analoger Ausgang mit Pulsweitenmodulation (PWM), spannungsgeregelt

→ Kapitel *Mögliche Betriebsarten E/A-Modul* (→ Seite 262)

► Die Konfiguration jedes einzelnen Ausgangs erfolgt über die Steuerungskonfiguration:
→ Kapitel *Ausgänge des integrierten E/A-Moduls konfigurieren* (→ Seite 259)

► [!] Zu den Grenzwerten unbedingt das Datenblatt beachten!

E/A-Modul Ausgangsgruppe Q3 (OUT10...OUT11)

15810

Bei diesen Ausgängen handelt es sich um eine Gruppe von Multifunktionskanälen.

Jeder einzelne dieser Ausgänge ist wahlweise wie folgt konfigurierbar:
- binärer Ausgang, plus-schaltend (BH)
- analoger Ausgang mit Pulsweitenmodulation (PWM)

→ Kapitel *Mögliche Betriebsarten E/A-Modul* (→ Seite 262)

► Die Konfiguration jedes einzelnen Ausgangs erfolgt über die Steuerungskonfiguration:
→ Kapitel *Ausgänge des integrierten E/A-Moduls konfigurieren* (→ Seite 259)

► [!] Zu den Grenzwerten unbedingt das Datenblatt beachten!

E/A-Modul Ausgangsgruppe Q4 (OUT12...OUT15)

15811

Bei diesen Ausgängen handelt es sich um eine Gruppe von Kanälen mit fest eingestellter Funktion.

Diese Ausgänge sind fix eingestellt wie folgt:
- binärer Ausgang, plus-schaltend (BH)

→ Kapitel *Mögliche Betriebsarten E/A-Modul* (→ Seite 262)

► [!] Zu den Grenzwerten unbedingt das Datenblatt beachten!

Schnittstellen-Beschreibung E/A-Modul

Inhalt

16426

CAN-Schnittstellen E/A-Modul

Inhalt

16608

Anschlüsse und Daten → Datenblatt

CAN: Schnittstellen und Protokolle: E/A-Modul in CR0133

15833
15835

Im integrierten E/A-Modul des Geräts sind folgende CAN-Schnittstellen und CAN-Protokolle verfügbar:

CAN-Schnittstelle	CAN 1	CAN 2	CAN 3	CAN 4
voreingestellte Download-ID	ID 123	ID 122	---	---
CAN-Protokolle	---	CANopen-Slave	---	---

Standard-Baudrate = 125 kBit/s

CAN: Schnittstellen und Protokolle: E/A-Modul in CR2532

16429
16435

Im integrierten E/A-Modul des Geräts sind folgende CAN-Schnittstellen und CAN-Protokolle verfügbar:

CAN-Schnittstelle	CAN 1	CAN 2	CAN 3	CAN 4
voreingestellte Download-ID	ID 125	ID 124	---	---
CAN-Protokolle	---	CANopen-Slave	---	---

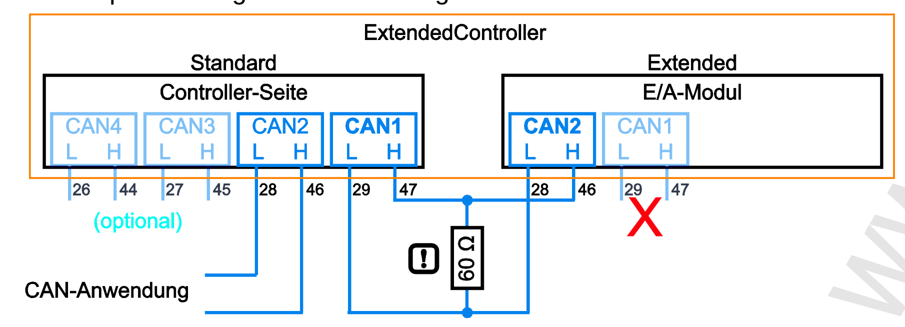
Standard-Baudrate = 250 kBit/s

Integriertes E/A-Modul ExB01 als CANopen-Slave anschließen

15829

Das integrierte E/A-Modul des Geräts basiert auf dem SmartController CR2530:
• diese Seite ist voreingestellt als CANopen-Slave ExB01
• diese Seite als Ein-/Ausgangsmodul behandeln!

Wir empfehlen folgende Verbindungsmethode:



- CAN1 des E/A-Moduls dient ausschließlich als Service- oder Maintenance-Schnittstelle!

► Für die Verbindung der Standard-Seite des Controllers mit dem integrierten E/A-Modul nur die gezeigte Verbindung nutzen!
Diese Anschlüsse NICHT für andere Zwecke verwenden!

► Für das CAN-Netzwerk in der Anwendung nur die Schnittstellen ≥ CAN2 der Standard-Seite nutzen!

7.3.2 Konfiguration des E/A-Moduls

Inhalt

16427

Programmiersystem einrichten (E/A-Modul)

Inhalt

16609

Programmiersystem manuell einrichten (E/A-Modul)

Inhalt

16610

Integriertes EA-Modul ExB01 einbinden

15828

❗ Das integrierte E/A-Modul des Geräts via CODESYS-Steuerungskonfiguration als CANopen-Slave einbinden!
Dies erfolgt mit der gleichen Methode wie beim Einbinden eines externen EA-Moduls:

► In der CODESYS-Steuerungskonfiguration die oberste Zeile (`CR0033 Configuration Vxx`) mit Links-Klick markieren.
► Mit Rechts-Klick das Kontext-Menü aufrufen.
► Dort [`Unterelement anhängen`] wählen.
► Im Auswahlmenü [`CANopen Master...`] wählen.
  ℹ Es ist immer sinnvoll, an CAN1 den ersten CANopen-Master zu konfigurieren.
► Mit Rechts-Klick auf [`CANopen-Master`] erneut das Kontext-Menü aufrufen.
► Dort [`Unterelement anhängen`] wählen.
► Im Auswahlmenü die EDS-Datei für das integrierte E/A-Modul des Geräts wählen: [`ExB01_Vxxyyzz.EDS`].

> Ergebnis:

```
⊟····CANopen Master[VAR]
    ⊟····ExB01 (EDS)  [VAR]
        ⊞····%QB256 Can-Output
        ⊞····%IB256 Can-Input
```

❗ Die IEC-Adressen für CAN-Input und CAN-Output ergeben sich aus folgenden Details:
• Typ des als CANopen-Master eingesetzten Geräts,
• Position des E/A-Moduls nach dem CANopen-Master,
• zugewiesene Node-ID.
❗ Das E/A-Modul belegt 3 aufeinander folgende Node-IDs. Regel:
⇨ [Node-ID des folgenden CAN-Slaves] ≥ [Node-ID des E/A-Moduls] + 3

► CAN-Parameter festlegen:
• Node-ID
• Nodeguarding
• Heartbeat-Settings

► Parametrieren der Ein- und Ausgänge im E/A-Modul:
→ Kapitel *Objektverzeichnis des integrierten E/A-Moduls* (→ Seite 266)

Programmiersystem über Templates einrichten (E/A-Modul)

16611
13745

ifm bietet vorgefertigte Templates (Programm-Vorlagen), womit Sie das Programmiersystem schnell, einfach und vollständig einrichten können.

970

> ℹ Beim Installieren der **ecomat*mobile***-DVD "Software, tools and documentation" wurden auch Projekte mit Vorlagen auf Ihrem Computer im Programmverzeichnis abgelegt:
> `…\ifm electronic\CoDeSys V…\Projects\Template_DVD_V…`
>
> ► Die gewünschte dort gespeicherte Vorlage in CODESYS öffnen mit:
> [Datei] > [Neu aus Vorlage…]
>
> > CODESYS legt ein neues Projekt an, dem der prinzipielle Programmaufbau entnommen werden kann. Es wird dringend empfohlen, dem gezeigten Schema zu folgen.

Funktionskonfiguration der Ein- und Ausgänge im E/A-Modul

Inhalt

16430

Eingänge des integrierten E/A-Moduls konfigurieren

Inhalt

16244

Software-Filter der Eingänge konfigurieren (E/A-Modul)

15898

Der Software-Filter ist fix eingestellt und nicht änderbar:

Tabelle: Grenzfrequenz Software-Tiefpassfilter am Analogeingang

FILTER	Filterfrequenz [Hz]	Sprungantwort [ms] für ...			Hinweise
		0…70 %	0…90 %	0…99 %	
fix	10	19	36	72	

Analogeingänge: Konfiguration und Diagnose (E/A-Modul ExB01)

15894

► Die Konfiguration jedes einzelnen Eingangs erfolgt über die Steuerungskonfiguration:
- unterhalb von [CANopen Master] auf die Zeile [ExB01 (EDS)] klicken
- Reiter [Service Data Objects] klicken
- Index / Sub-Index des gewünschten Parameters wählen
- in Spalte [Value] auf den bestehenden Wert klicken
- Wert ändern und mit [ENTER] bestätigen

zulässige Werte → Kapitel *Eingänge: Betriebsarten (E/A-Modul)* (→ Seite 264)

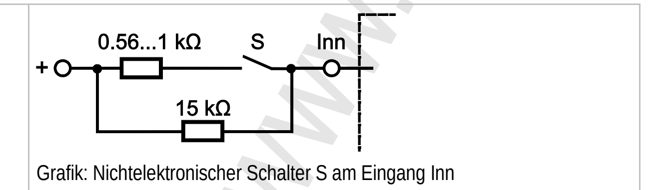
> Werden die Analogeingänge auf Strommessung konfiguriert, wird bei Überschreiten des Endwertes (23 mA für ≥ 40 ms) in den sicheren Spannungsmessbereich (0...32 V DC) geschaltet. Dies meldet das PDO1 als "overcurrent". Nach etwa einer Sekunde schaltet der Eingang selbsttätig auf den Strommessbereich zurück.

Binäreingänge: Konfiguration und Diagnose (E/A-Modul ExB01)

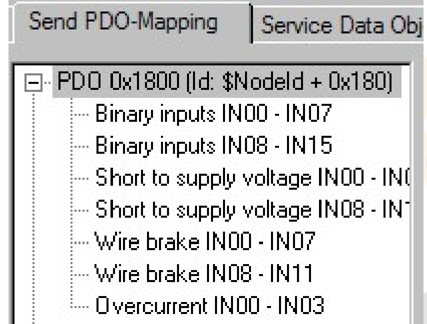
15896

► Die Konfiguration jedes einzelnen Eingangs erfolgt über die Steuerungskonfiguration:
- unterhalb von [CANopen Master] auf die Zeile [ExB01 (EDS)] klicken
- Reiter [Service Data Objects] klicken
- Index / Sub-Index des gewünschten Parameters wählen
- in Spalte [Value] auf den bestehenden Wert klicken
- Wert ändern und mit [ENTER] bestätigen

zulässige Werte → Kapitel *Eingänge: Betriebsarten (E/A-Modul)* (→ Seite 264)

NAMUR-Diagnose für digitale Signale bei nichtelektronischen Schaltern: ► Schalter mit einer zusätzlichen Widerstandsbeschaltung versehen!	 Grafik: Nichtelektronischer Schalter S am Eingang Inn

> Das Diagnose-Ergebnis zeigt PDO 1



Schnelle Eingänge: E/A-Modul ExB01

15869

Die Geräte verfügen über schnelle Zähl-/Impulseingänge für eine Eingangsfrequenz bis 30 kHz (→ Datenblatt).

❗ Werden z.B. mechanische Schalter an diesen Eingängen angeschlossen, kann es durch Kontaktprellen zu Fehlsignalen in der Steuerung kommen.

Die Konfiguration jedes einzelnen Eingangs erfolgt über die Steuerungskonfiguration:
zulässige Werte → Kapitel *Eingänge: Betriebsarten (E/A-Modul)* (→ Seite 264)

3804

Durch die zulässigen hohen Eingangsfrequenzen können auch Fehlsignale erkannt werden, z.B. prellende Kontakte mechanischer Schalter.

► Bei Bedarf die Fehlsignale im Anwendungsprogramm unterdrücken!

Ausgänge des integrierten E/A-Moduls konfigurieren

Inhalt

16248

Software-Filter der Ausgänge konfigurieren (E/A-Modul)

15900

Für das E/A-Modul gilt:
Der Software-Filter ist fix eingestellt und nicht änderbar.

Tabelle: Grenzfrequenz Software-Tiefpassfilter am PWM-Ausgang

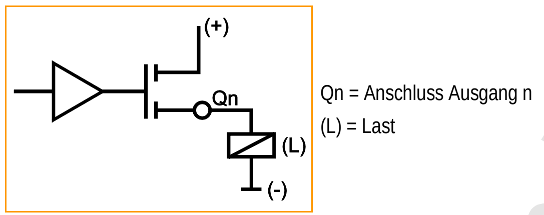
FILTER	Filterfrequenz [Hz]	Sprungantwort [ms] für …			Hinweise
		0…90 %	0…95 %	0…99 %	
fix	52	7,2	9,4	14,4	

Binärausgänge: Konfiguration und Diagnose (E/A-Modul ExB01)

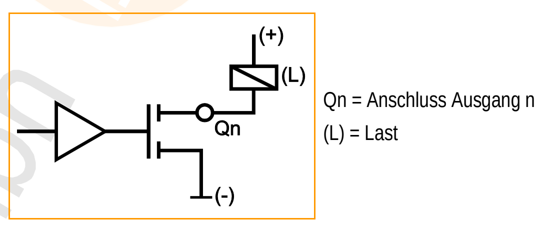
15882

Bei den Geräte-Ausgängen sind folgende Betriebsarten möglich (→ Datenblatt):
- binärer Ausgang, plus-schaltend (BH) mit/ohne Diagnosefunktion

15450



Prinzipschaltung Ausgang plus-schaltend (BH)
für positives Ausgangssignal



Prinzipschaltung Ausgang minus-schaltend (BL)
für negatives Ausgangssignal

13975

⚠ WARNUNG

Gefährlicher Wiederanlauf möglich!
Gefahr von Personenschaden! Gefahr von Sachschaden an der Maschine/Anlage!

Wird ein Ausgang im Fehlerfall hardwaremäßig abgeschaltet, ändert sich der durch das Anwendungsprogramm erzeugte logische Zustand dadurch nicht.

▶ Abhilfe:
- Die Ausgänge zunächst im Anwendungsprogramm logisch zurücksetzen!
- Fehler beseitigen!
- Ausgänge situationsabhängig wieder setzen.

Binärausgänge: Konfiguration (E/A-Modul ExB01)

15887

► Die Konfiguration jedes einzelnen Ausgangs erfolgt über die Steuerungskonfiguration:
- unterhalb von [CANopen Master] auf die Zeile [ExB01 (EDS)] klicken
- Reiter [Service Data Objects] klicken
- Index / Sub-Index des gewünschten Parameters wählen
- in Spalte [Value] auf den bestehenden Wert klicken
- Wert ändern und mit [ENTER] bestätigen

zulässige Werte → Kapitel *Ausgänge: Betriebsarten (E/A-Modul)* (→ Seite 265)

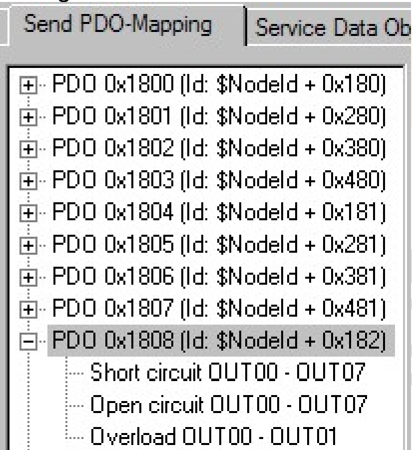
Binärausgänge: Diagnose (E/A-Modul ExB01)

15889

► Die Konfiguration jedes einzelnen Ausgangs erfolgt über die Steuerungskonfiguration:
Aktivieren der Diagnose mit...
- Modus = 15 (OUT_BINARY_HIGH_DIAG) oder
- Modus = 16 (OUT_BINARY_HIGH_DIAG_PROT)

zulässige Werte → Kapitel *Ausgänge: Betriebsarten (E/A-Modul)* (→ Seite 265)

> Das Ergebnis zeigt PDO 9:

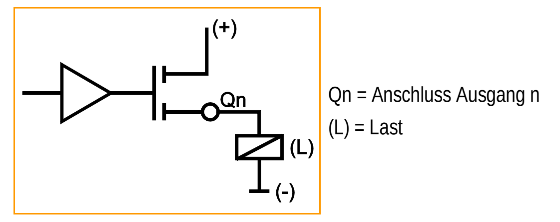


PWM-Ausgänge: E/A-Modul ExB01

16415

Bei den Geräte-Ausgängen sind folgende Betriebsarten möglich (→ Datenblatt):
• PWM-Ausgang, plus-schaltend (BH) ohne Diagnosefunktion

15451



Qn = Anschluss Ausgang n
(L) = Last

Prinzipschaltung Ausgang plus-schaltend (BH)
für positives Ausgangssignal

16253

⚠ WARNUNG

Sach- oder Körperschäden möglich durch Fehlfunktionen!

Für Ausgänge im PWM-Modus gilt:
• es gibt keine Diagnosefunktionen

9980

HINWEIS

PWM-Ausgänge dürfen NICHT parallel betrieben werden, um z.B. den max. Ausgangsstrom zu erhöhen. Die Ausgänge arbeiten nicht synchron.

Andernfalls kann die komplette Last über nur einen Ausgang gehen. Die Strommessung funktioniert dann nicht mehr.

- PWM-Ausgänge können mit und ohne Stromregelfunktion betrieben werden.
  Stromgeregelte PWM-Ausgänge werden überwiegend zur Ansteuerung von proportionalen Hydraulikfunktionen genutzt.

Verfügbarkeit von PWM

16364

Gerät	Anzahl verfügbare PWM-Ausgänge	davon stromgeregelt (PWMi)	PWM-Frequenz [Hz]
integriertes E/A-Modul ExB01	12	2	20...250

Ausgänge konfigurieren für PWM-Funktionen

15888

Für die PWM-Funktion der Ausgänge stehen folgende Einstellungen zur Verfügung:
• Modus = 4 (OUT_PWM) oder
• Modus = 5 (OUT_CURRENT)
zulässige Werte → Kapitel *Ausgänge: Betriebsarten (E/A-Modul)* (→ Seite 265)

Stromregelung mit PWM (= PWMi)

14722

Über die im Controller integrierten Strommesskanäle kann eine Strommessung des Spulenstroms durchgeführt werden. Dadurch kann zum Beispiel der Strom bei einer Spulenerwärmung nachgeregelt werden. Damit bleiben die Hydraulikverhältnisse im System gleich.

Grundsätzlich sind die stromgeregelten Ausgänge gegen Kurzschluss geschützt.

Mögliche Betriebsarten E/A-Modul

Inhalt

16440

Übersicht

15859

Für die Ein- und Ausgänge sind folgende Betriebsarten möglich (konkret: → folgende Seiten):

SDO-Wert dez	hex	Modus	Eingang Ausgang	Beschreibung
0	0x00	OFF	Eingang Ausgang	ausgeschaltet, ohne Funktion
1	0x01	IN_BINARY_LOW_DIGITAL	Eingang	binär plus-schaltend, digital erfasst (möglichst statt dessen Modus 10 verwenden!)
2	0x02	OUT_BINARY_HIGH	Ausgang	binär plus-schaltend: Ausgang = FALSE ⇨ 0 V Ausgang = TRUE ⇨ Versorgungsspannung der Ausgänge
3	0x03	IN_VOLTAGE_10V	Eingang	analoge Spannungsmessung im 10-V-Messbereich
4	0x04	OUT_PWM	Ausgang	PWM-Betrieb
5	0x05	OUT_CURRENT	Ausgang	stromgeregelt
6	0x06	IN_VOLTAGE_RATIO	Eingang	analoge Spannungsmessung ratiometrisch zur Versorgungsspannung VBBS
7	0x07	IN_CURRENT	Eingang	analoge Strommessung (bis 23 mA)
8	0x08	---	---	reserviert
9	0x09	IN_VOLTAGE_32	Eingang	analoge Spannungsmessung im 32-V-Messbereich
10	0x0A	IN_BINARY_LOW	Eingang	binär plus-schaltend (abhängig von der Versorgungsspannung VBBS) (analog oder digital erfasst)
11	0x0B	IN_BINARY_LOW_DIAG	Eingang	binär plus-schaltend mit Diagnose (analog erfasst) abhängig von der Versorgungsspannung VBBS Diagnose auf Schluss gegen VBBS oder GND
12	0x0C	IN_BINARY_HIGH	Eingang	binär minus-schaltend (analog erfasst) abhängig von der Versorgungsspannung VBBS
13	0x0D	---	---	reserviert
14	0x0E	IN_FREQUENCY	Eingang	Frequenzmessung (digital erfasst)
15	0x0F	OUT_BINARY_HIGH_DIAG	Ausgang	binär plus-schaltend: Ausgang = FALSE ⇨ 0 V Ausgang = TRUE ⇨ Versorgungsspannung der Ausgänge Diagnose auf Leiterbruch und Kurzschluss
16	0x10	OUT_BINARY_HIGH_DIAG_PROT	Ausgang	binär plus-schaltend: Ausgang = FALSE ⇨ 0 V Ausgang = TRUE ⇨ Versorgungsspannung der Ausgänge Diagnose auf Leiterbruch und Kurzschluss Abschaltung des Ausgangs bei Kurzschluss
17	0x11	---	---	reserviert
18	0x12	IN_RESISTOR	Eingang	Widerstandsmessung (analog erfasst)
19	0x13	---	---	reserviert
20	0x14	IN_PERIOD_RATIO	Eingang	Periodendauermessung als Ratio-Verhältnis (digital erfasst)
21	0x15	---	---	reserviert
22	0x16	---	---	reserviert
23	0x17	---	---	reserviert
24	0x18	---	---	reserviert

Eingänge: Betriebsarten (E/A-Modul)

15965

► Die Konfiguration jedes einzelnen Eingangs erfolgt über die Steuerungskonfiguration:
- unterhalb von [CANopen Master] auf die Zeile [ExB01 (EDS)] klicken
- Reiter [Service Data Objects] klicken
- Index / Sub-Index des gewünschten Parameters wählen
- in Spalte [Value] auf den bestehenden Wert klicken
- Wert ändern und mit [ENTER] bestätigen

= diese Konfiguration ist voreingestellt

Eingänge	mögliche Betriebsart		im Objektverzeichnis Index	Sub-Index	Wert	
					dez	hex
IN00...IN03	Aus		0x2000	0x01...0x04	0	0x00
	Spannungseingang	0...10 000 mV	0x2000	0x01...0x04	3	0x03
	Spannungseingang ratiometrisch	0...1 000 ‰	0x2000	0x01...0x04	6	0x06
	Stromeingang	0...20 000 µA	0x2000	0x01...0x04	7	0x07
	Spannungseingang	0...32 000 mV	0x2000	0x01...0x04	9	0x09
	binärer Eingang	plus-schaltend	0x2000	0x01...0x04	10	0x0A
	binärer Eingang mit Diagnose (Namur)	plus-schaltend	0x2000	0x01...0x04	11	0x0B
	binärer Eingang	minus-schaltend	0x2000	0x01...0x04	12	0x0C
IN04...IN05	Aus		0x2000	0x05...0x06	0	0x00
	binärer Eingang	plus-schaltend	0x2000	0x05...0x06	10	0x0A
	binärer Eingang mit Diagnose (Namur)	plus-schaltend	0x2000	0x05...0x06	11	0x0B
	Widerstandseingang	16...30 000 Ohm	0x2000	0x05...0x06	18	0x12
IN06...IN11	Aus		0x2000	0x07...0x0C	0	0x00
	binärer Eingang	plus-schaltend	0x2000	0x07...0x0C	10	0x0A
	binärer Eingang mit Diagnose (Namur)	plus-schaltend	0x2000	0x07...0x0C	11	0x0B
IN12...IN15	Aus		0x2000	0x0D...0x10	0	0x00
	binärer Eingang digital ausgewertet	plus-schaltend	0x2000	0x0D...0x10	1	0x01
	Frequenzmessung	0...30 000 Hz	0x2000	0x0D...0x10	14	0x0E
	Periodendauer- und Ratiomessung	0,1...3 000 Hz	0x2000	0x0D...0x10	20	0x14

Ausgänge: Betriebsarten (E/A-Modul)

15966

► Die Konfiguration jedes einzelnen Ausgangs erfolgt über die Steuerungskonfiguration:
- unterhalb von [CANopen Master] auf die Zeile [ExB01 (EDS)] klicken
- Reiter [Service Data Objects] klicken
- Index / Sub-Index des gewünschten Parameters wählen
- in Spalte [Value] auf den bestehenden Wert klicken
- Wert ändern und mit [ENTER] bestätigen

= diese Konfiguration ist voreingestellt

Ausgänge	mögliche Betriebsart		im Objektverzeichnis Index	Sub-Index	Wert	
					dez	hex
OUT00 ...OUT01	Aus		0x2000	0x11...0x12	0	0x00
	binärer Ausgang	plus-schaltend	0x2000	0x11...0x12	2	0x02
	analoger Ausgang mit Pulsweitenmodulation		0x2000	0x11...0x12	4	0x04
	analoger stromgeregelter Ausgang		0x2000	0x11...0x12	5	0x05
	binärer Ausgang mit Diagnose	plus-schaltend	0x2000	0x11...0x12	15	0x0F
	binärer Ausgang mit Diagnose und Protection	plus-schaltend	0x2000	0x11...0x12	16	0x10
OUT02 ...OUT07	Aus		0x2000	0x13...0x18	0	0x00
	binärer Ausgang	plus-schaltend	0x2000	0x13...0x18	2	0x02
	analoger Ausgang mit Pulsweitenmodulation		0x2000	0x13...0x18	4	0x04
	binärer Ausgang mit Diagnose	plus-schaltend	0x2000	0x13...0x18	15	0x0F
	binärer Ausgang mit Diagnose und Protection	plus-schaltend	0x2000	0x13...0x18	16	0x10
OUT08 ...OUT09	Aus		0x2000	0x19...0x1A	0	0x00
	binärer Ausgang	plus-schaltend	0x2000	0x19...0x1A	2	0x02
	analoger Ausgang mit Pulsweitenmodulation		0x2000	0x19...0x1A	4	0x04
	analoger Ausgang mit Pulsweitenmodulation, spannungsgeregelt	(an Pins 25 + 43)	0x2000	0x19...0x1A	4	0x04
OUT10 ...OUT11	Aus		0x2000	0x1B...0x1C	0	0x00
	binärer Ausgang	plus-schaltend	0x2000	0x1B...0x1C	2	0x02
	analoger Ausgang mit Pulsweitenmodulation		0x2000	0x1B...0x1C	4	0x04
OUT12 ...OUT15	Aus		0x2000	0x1D...0x20	0	0x00
	binärer Ausgang	plus-schaltend	0x2000	0x1D...0x20	2	0x02

7.3.3 Objektverzeichnis des integrierten E/A-Moduls

Inhalt

15837

Objektverzeichnis-Parametertabellen, Übersicht

Inhalt

15977

Allgemein

15967

► Das automatische Sichern der Kommunikations- und Geräteparameter über den Eintrag [Save Parameter] (→ Objektverzeichnis, Index 0x1010) aus- oder einschalten:

- Wenn SubIndex 0x1 = 0x02:
  alle Parameter bei einer Änderung automatisch sichern.
- Wenn SubIndex 0x1 = 0x00:
  die Parameter nicht automatisch sichern.
  Geänderte Parameter sind nur gültig ...
  • bis zum Ausschalten des Geräts
  • bis zum nächsten Reset.

► Über die Funktion [Restore] (→ Objektverzeichnis, Index 0x1011) die Parameter mit den voreingestellten Werten belegen (gilt nicht für Baudrate und Node-ID). Diese Werte sind gültig nach dem nächsten Einschalten der Versorgungsspannung.

Datentypen in der EDS-Datei

16409

EDS-Datentyp	IEC-Datentyp	min. Wert	max. Wert	Größe im Speicher
	BOOL	FALSE	TRUE	8 Bit = 1 Byte
	BYTE	0	255	8 Bit = 1 Byte
	WORD	0	65 535	16 Bit = 2 Bytes
	DWORD	0	4 294 967 295	32 Bit = 4 Bytes
	SINT	-128	127	8 Bit = 1 Byte
0x0005	USINT	0	255	8 Bit = 1 Byte
0x0003	INT	-32 768	32 767	16 Bit = 2 Bytes
0x0006	UINT	0	65 535	16 Bit = 2 Bytes
	DINT	-2 147 483 648	2 147 483 647	32 Bit = 4 Bytes
0x0007	UDINT	0	4 294 967 295	32 Bit = 4 Bytes
0x0008	REAL	-3,402823466 • 1038	3,402823466 • 1038	32 Bit = 4 Bytes
	ULINT	0	18 446 744 073 709 551 615	64 Bit = 8 Bytes
0x0009	STRING			number of char. + 1

Objektverzeichnis Pflichtobjekte (Index 0x1000...0x1FFF), Übersicht

15979

Objektverzeichnis		Parameter-Beschreibung	Parameter für	Parameter-Wert voreingestellt	Änderung automatisch gesichert?	Änderung wann wirksam?
Index	Sub-Idx					
0x1000		Device type	device	0xF0191	ja	sofort (via CAN-Stack)
0x1001		Error register	device	---	ja	sofort (via CAN-Stack)
0x1018		Device identification	device	---	--	--
	0x1	Vendor ID	device	6907501	ja	einmalig bei Herstellung
	0x2	Product code	device	0	ja	einmalig bei Herstellung
	0x3	Revision number	device	0	ja	einmalig bei Herstellung
	0x4	Serial number	device	0	ja	einmalig bei Herstellung

Objektverzeichnis optionale Objekte (Index 0x1000...0x1FFF), Übersicht

15980

Objektverzeichnis		Parameter-Beschreibung	Parameter für	Parameter-Wert voreingestellt	Änderung automatisch gesichert?	Änderung wann wirksam?
Index	Sub-Idx					
0x1003	0x1…0x 5	Predefined error field	CANopen Basiskonfiguration	0	ja	sofort (via CAN-Stack)
0x1005		COB ID synch message	CANopen Basiskonfiguration	0x80	ja	sofort (via CAN-Stack)
0x1006		Communication cycle period	CANopen Basiskonfiguration	0	ja	sofort
0x1008		Manufacturer device name	CANopen Basiskonfiguration	ExB01	ja	sofort
0x1009		Manufacturer hardware version	CANopen Basiskonfiguration	V00.00.00	ja	sofort
0x100A		Manufacturer software version	CANopen Basiskonfiguration	V00.00.00	ja	sofort
0x100C		Guard time	CANopen Basiskonfiguration	0	ja	sofort
0x100D		Lifetime factor	CANopen Basiskonfiguration	0	ja	sofort
0x1010		Store parameters	CANopen Basiskonfiguration		ja	sofort
	0x1	Save all parameters	CANopen Basiskonfiguration	1	ja	sofort
0x1011		Restore default parameters	CANopen Basiskonfiguration		nein	nach Reset
	0x1	Restore all default parameters	CANopen Basiskonfiguration	1	nein	nach Reset
0x1014		COB ID emergency	CANopen Basiskonfiguration	0x80 + Node-ID	ja	sofort
0x1016		Consumer heartbeat times	CANopen Basiskonfiguration		--	--
	0x1	Consumer heartbeat time	CANopen Basiskonfiguration	0	ja	sofort
0 1017		Producer heartbeat time	CANopen Basiskonfiguration	0	ja	sofort
0 1400		Receive PDO communication parameter	Konfiguration Empfangs-PDO 1		--	--
	0x1	COB ID used by PDO	Konfiguration Empfangs-PDO 1	0x0200 + Node-ID	ja	nach PreOp
	0x2	transmission type	Konfiguration Empfangs-PDO 1	1	ja	sofort
0x1401		Receive PDO communication parameter	Konfiguration Empfangs-PDO 2		--	--
	0x1	COB ID used by PDO	Konfiguration Empfangs-PDO 2	0x0300 + Node-ID	ja	nach PreOp
	0x2	transmission type	Konfiguration Empfangs-PDO 2	1	ja	sofort
0x1402		Receive PDO communication parameter	Konfiguration Empfangs-PDO 3		--	--
	0x1	COB ID used by PDO	Konfiguration Empfangs-PDO 3	0x0400 + Node-ID	ja	nach PreOp

Objektverzeichnis		Parameter-Beschreibung	Parameter für	Parameter-Wert voreingestellt	Änderung automatisch gesichert?	Änderung wann wirksam?
Index	Sub-Idx					
	0x2	transmission type	Konfiguration Empfangs-PDO 3	1	ja	sofort
0x1403		Receive PDO communication parameter	Konfiguration Empfangs-PDO 4		--	--
	0x1	COB ID used by PDO	Konfiguration Empfangs-PDO 4	0x0500 + Node-ID	ja	nach PreOp
	0x2	transmission type	Konfiguration Empfangs-PDO 4	1	ja	sofort
0x1600		Receive PDO mapping	Mapping Empfangs-PDO 1		ja	nach PreOp
	0x1	PDO mapping	Mapping Empfangs-PDO 1	0x6200 0108	ja	nach PreOp
	0x2	PDO mapping	Mapping Empfangs-PDO 1	0x6200 0208	ja	nach PreOp
0x1601		Receive PDO mapping	Mapping Empfangs-PDO 2		ja	nach PreOp
	0x1	PDO mapping	Mapping Empfangs-PDO 2	0x6414 0110	ja	nach PreOp
	0x2	PDO mapping	Mapping Empfangs-PDO 2	0x6414 0210	ja	nach PreOp
	0x3	PDO mapping	Mapping Empfangs-PDO 2	0x6414 0310	ja	nach PreOp
	0x4	PDO mapping	Mapping Empfangs-PDO 2	0x6414 0410	ja	nach PreOp
0x1602		Receive PDO mapping	Mapping Empfangs-PDO 3		ja	nach PreOp
	0x1	PDO mapping	Mapping Empfangs-PDO 3	0x6414 0510	ja	nach PreOp
	0x2	PDO mapping	Mapping Empfangs-PDO 3	0x6414 0610	ja	nach PreOp
	0x3	PDO mapping	Mapping Empfangs-PDO 3	0x6414 0710	ja	nach PreOp
	0x4	PDO mapping	Mapping Empfangs-PDO 3	0x6414 0810	ja	nach PreOp
0x1603		Receive PDO mapping	Mapping Empfangs-PDO 4		ja	nach PreOp
	0x1	PDO mapping	Mapping Empfangs-PDO 4	0x6414 0910	ja	nach PreOp
	0x2	PDO mapping	Mapping Empfangs-PDO 4	0x6414 0A10	ja	nach PreOp
	0x3	PDO mapping	Mapping Empfangs-PDO 4	0x6414 0B10	ja	nach PreOp
	0x4	PDO mapping	Mapping Empfangs-PDO 4	0x6414 0C10	ja	nach PreOp
	0x5	PDO mapping	Mapping Empfangs-PDO 4	0x00	ja	nach PreOp
0x1800		Transmit PDO communication parameter	Konfiguration Sende-PDO 1		--	--
	0x1	COB ID used by PDO	Konfiguration Sende-PDO 1	0x180 + Node-ID	--	--

Objektverzeichnis		Parameter-Beschreibung	Parameter für	Parameter-Wert voreingestellt	Änderung automatisch gesichert?	Änderung wann wirksam?
Index	Sub-Idx					
	0x2	transmission type	Konfiguration Sende-PDO 1	1	ja	sofort
	0x3	inhibit time	Konfiguration Sende-PDO 1	0	ja	sofort
	0x4	reserved	Konfiguration Sende-PDO 1	0	nein	--
	0x5	event time	Konfiguration Sende-PDO 1	0	ja	sofort
0x1801		Transmit PDO communication parameter	Konfiguration Sende-PDO 2		--	--
	0x1	COB ID used by PDO	Konfiguration Sende-PDO 2	0x280 + Node-ID	--	--
	0x2	transmission type	Konfiguration Sende-PDO 2	1	ja	sofort
	0x3	inhibit time	Konfiguration Sende-PDO 2	0	ja	sofort
	0x4	reserved	Konfiguration Sende-PDO 2	0	nein	--
	0x5	event time	Konfiguration Sende-PDO 2	0	ja	sofort
0x1802		Transmit PDO communication parameter	Konfiguration Sende-PDO 3		--	--
	0x1	COB ID used by PDO	Konfiguration Sende-PDO 3	0x380 + Node-ID	--	--
	0x2	transmission type	Konfiguration Sende-PDO 3	1	ja	sofort
	0x3	inhibit time	Konfiguration Sende-PDO 3	0	ja	sofort
	0x4	reserved	Konfiguration Sende-PDO 3	0	nein	--
	0x5	event time	Konfiguration Sende-PDO 3	0	ja	sofort
0x1803		Transmit PDO communication parameter	Konfiguration Sende-PDO 4		--	--
	0x1	COB ID used by PDO	Konfiguration Sende-PDO 4	0x480 + Node-ID	--	--
	0x2	transmission type	Konfiguration Sende-PDO 4	1	ja	sofort
	0x3	inhibit time	Konfiguration Sende-PDO 4	0	ja	sofort
	0x4	reserved	Konfiguration Sende-PDO 4	0	nein	--
	0x5	event time	Konfiguration Sende-PDO 4	0	ja	sofort
0x1804		Transmit PDO communication parameter	Konfiguration Sende-PDO 5		--	--
	0x1	COB ID used by PDO	Konfiguration Sende-PDO 5	0x181 + Node-ID	--	--
	0x2	transmission type	Konfiguration Sende-PDO 5	1	ja	sofort
	0x3	inhibit time	Konfiguration Sende-PDO 5	0	ja	sofort

Objektverzeichnis		Parameter-Beschreibung	Parameter für	Parameter-Wert voreingestellt	Änderung automatisch gesichert?	Änderung wann wirksam?
Index	Sub-Idx					
	0x4	reserved	Konfiguration Sende-PDO 5	0	nein	--
	0x5	event time	Konfiguration Sende-PDO 5	0	ja	sofort
0x1805		Transmit PDO communication parameter	Konfiguration Sende-PDO 6		--	--
	0x1	COB ID used by PDO	Konfiguration Sende-PDO 6	0x281 + Node-ID	--	--
	0x2	transmission type	Konfiguration Sende-PDO 6	1	ja	sofort
	0x3	inhibit time	Konfiguration Sende-PDO 6	0	ja	sofort
	0x4	reserved	Konfiguration Sende-PDO 6	0	nein	--
	0x5	event time	Konfiguration Sende-PDO 6	0	ja	sofort
0x1806		Transmit PDO communication parameter	Konfiguration Sende-PDO 7		--	--
	0x1	COB ID used by PDO	Konfiguration Sende-PDO 7	0x381 + Node-ID	--	--
	0x2	transmission type	Konfiguration Sende-PDO 7	1	ja	sofort
	0x3	inhibit time	Konfiguration Sende-PDO 7	0	ja	sofort
	0x4	reserved	Konfiguration Sende-PDO 7	0	nein	--
	0x5	event time	Konfiguration Sende-PDO 7	0	ja	sofort
0x1807		Transmit PDO communication parameter	Konfiguration Sende-PDO 8		--	--
	0x1	COB ID used by PDO	Konfiguration Sende-PDO 8	0x481 + Node-ID	--	--
	0x2	transmission type	Konfiguration Sende-PDO 8	1	ja	sofort
	0x3	inhibit time	Konfiguration Sende-PDO 8	0	ja	sofort
	0x4	reserved	Konfiguration Sende-PDO 8	0	nein	--
	0x5	event time	Konfiguration Sende-PDO 8	0	ja	sofort
0x1808		Transmit PDO communication parameter	Konfiguration Sende-PDO 9		--	--
	0x1	COB ID used by PDO	Konfiguration Sende-PDO 9	0x182 + Node-ID	--	--
	0x2	transmission type	Konfiguration Sende-PDO 9	1	ja	sofort
	0x3	inhibit time	Konfiguration Sende-PDO 9	0	ja	sofort
	0x4	reserved	Konfiguration Sende-PDO 9	0	nein	--
	0x5	event time	Konfiguration Sende-PDO 9	0	ja	sofort

Objektverzeichnis		Parameter-Beschreibung	Parameter für	Parameter-Wert voreingestellt	Änderung automatisch gesichert?	Änderung wann wirksam?
Index	Sub-Idx					
0x1809		Transmit PDO communication parameter	Konfiguration Sende-PDO 10		--	--
	0x1	COB ID used by PDO	Konfiguration Sende-PDO 10	0x282 + Node-ID	--	--
	0x2	transmission type	Konfiguration Sende-PDO 10	1	ja	sofort
	0x3	inhibit time	Konfiguration Sende-PDO 10	0	ja	sofort
	0x4	reserved	Konfiguration Sende-PDO 10	0	nein	--
	0x5	event time	Konfiguration Sende-PDO 10	0	ja	sofort
0x1A00		Transmit PDO mapping	Mapping Sende-PDO 1		ja	nach PreOp
	0x1	PDO mapping	Mapping Sende-PDO 1	0x6000 0108	ja	nach PreOp
	0x2	PDO mapping	Mapping Sende-PDO 1	0x6000 0208	ja	nach PreOp
	0x3	PDO mapping	Mapping Sende-PDO 1	0x2020 0108	ja	nach PreOp
	0x4	PDO mapping	Mapping Sende-PDO 1	0x2020 0208	ja	nach PreOp
	0x5	PDO mapping	Mapping Sende-PDO 1	0x2021 0108	ja	nach PreOp
	0x6	PDO mapping	Mapping Sende-PDO 1	0x2021 0208	ja	nach PreOp
	0x7	PDO mapping	Mapping Sende-PDO 1	0x2025 0108	ja	nach PreOp
0x1A01		Transmit PDO mapping	Mapping Sende-PDO 2		ja	nach PreOp
	0x1	PDO mapping	Mapping Sende-PDO 2	0x6404 0110	ja	nach PreOp
	0x2	PDO mapping	Mapping Sende-PDO 2	0x6404 0210	ja	nach PreOp
	0x3	PDO mapping	Mapping Sende-PDO 2	0x6404 0310	ja	nach PreOp
	0x4	PDO mapping	Mapping Sende-PDO 2	0x6404 0410	ja	nach PreOp
0x1A02		Transmit PDO mapping	Mapping Sende-PDO 3		ja	nach PreOp
	0x1	PDO mapping	Mapping Sende-PDO 3	0x2030 0110	ja	nach PreOp
	0x2	PDO mapping	Mapping Sende-PDO 3	0x2030 0210	ja	nach PreOp
	0x3	PDO mapping	Mapping Sende-PDO 3	0x2002 0110	ja	nach PreOp
	0x4	PDO mapping	Mapping Sende-PDO 3	0x2002 0210	ja	nach PreOp
	0x5	PDO mapping	Mapping Sende-PDO 3	0	ja	nach PreOp

Objektverzeichnis		Parameter-Beschreibung	Parameter für	Parameter-Wert voreingestellt	Änderung automatisch gesichert?	Änderung wann wirksam?
Index	Sub-Idx					
0x1A03		Transmit PDO mapping	Mapping Sende-PDO 4		ja	nach PreOp
	0x1	PDO mapping	Mapping Sende-PDO 4	0x2012 0120	ja	nach PreOp
	0x2	PDO mapping	Mapping Sende-PDO 4	0x2012 0220	ja	nach PreOp
	0x3	PDO mapping	Mapping Sende-PDO 4	0	ja	nach PreOp
0x1A04		Transmit PDO mapping	Mapping Sende-PDO 5		ja	nach PreOp
	0x1	PDO mapping	Mapping Sende-PDO 5	0x2012 0320	ja	nach PreOp
	0x2	PDO mapping	Mapping Sende-PDO 5	0x2012 0420	ja	nach PreOp
	0x3	PDO mapping	Mapping Sende-PDO 5	0	ja	nach PreOp
0x1A05		Transmit PDO mapping	Mapping Sende-PDO 6		ja	nach PreOp
	0x1	PDO mapping	Mapping Sende-PDO 6	0x2014 0110	ja	nach PreOp
	0x2	PDO mapping	Mapping Sende-PDO 6	0x2014 0210	ja	nach PreOp
	0x3	PDO mapping	Mapping Sende-PDO 6	0x2014 0310	ja	nach PreOp
	0x4	PDO mapping	Mapping Sende-PDO 6	0x2014 0410	ja	nach PreOp
	0x5	PDO mapping	Mapping Sende-PDO 6	0	ja	nach PreOp
0x1A06		Transmit PDO mapping	Mapping Sende-PDO 7		ja	nach PreOp
	0x1	PDO mapping	Mapping Sende-PDO 7	0x2015 0120	ja	nach PreOp
	0x2	PDO mapping	Mapping Sende-PDO 7	0x2015 0220	ja	nach PreOp
	0x3	PDO mapping	Mapping Sende-PDO 7	0	ja	nach PreOp
0x1A07		Transmit PDO mapping	Mapping Sende-PDO 8		ja	nach PreOp
	0x1	PDO mapping	Mapping Sende-PDO 8	0x2015 0320	ja	nach PreOp
	0x2	PDO mapping	Mapping Sende-PDO 8	0x2015 0420	ja	nach PreOp
	0x3	PDO mapping	Mapping Sende-PDO 8	0	ja	nach PreOp
0x1A08		Transmit PDO mapping	Mapping Sende-PDO 9		ja	nach PreOp
	0x1	PDO mapping	Mapping Sende-PDO 9	0x2022 0108	ja	nach PreOp
	0x2	PDO mapping	Mapping Sende-PDO 9	0x2023 0108	ja	nach PreOp
	0x3	PDO mapping	Mapping Sende-PDO 9	0x2024 0108	ja	nach PreOp

Objektverzeichnis		Parameter-Beschreibung	Parameter für	Parameter-Wert voreingestellt	Änderung automatisch gesichert?	Änderung wann wirksam?
Index	Sub-Idx					
	0x4	PDO mapping	Mapping Sende-PDO 9	0	ja	nach PreOp
0x1A09		Transmit PDO mapping	Mapping Sende-PDO 10		ja	nach PreOp
	0x1	PDO mapping	Mapping Sende-PDO 10	0x2040 0110	ja	nach PreOp
	0x2	PDO mapping	Mapping Sende-PDO 10	0x2041 0110	ja	nach PreOp
	0x3	PDO mapping	Mapping Sende-PDO 10	0x2041 0210	ja	nach PreOp
	0x4	PDO mapping	Mapping Sende-PDO 10	0x2050 0010	ja	nach PreOp
	0x5	PDO mapping	Mapping Sende-PDO 10	0	ja	nach PreOp

Life Time Factor 0 wird als 1 interpretiert.

Das erste Guardprotokoll wird als "Start Guarding" gewertet, auch wenn zu diesem Zeitpunkt das Guarding noch nicht aktiviert ist (Guardtime = 0).

Objektverzeichnis herstellerspezifische Objekte (Index 0x2000...0x6FFF), Übersicht

15978

Objektverzeichnis Index	Parameter-Beschreibung	Parameter für	Parameter-Wert voreingestellt	Änderung automatisch gesichert?	Änderung wann wirksam?
0x2000	I/O configuration	IN00…IN11 IN12…IN15	10 01	ja	nach PreOp
0x2001	PWM frequency	OUT00…OUT11	100	ja	nach PreOp
0x2002	Current value	OUT00…OUT01	0	ja	nach PreOp
0x2004	P-value	OUT00…OUT01	30	ja	nach PreOp
0x2005	I-value	OUT00…OUT01	20	ja	nach PreOp
0x2006	PWM dither frequency	OUT00…OUT11	0	ja	nach PreOp
0x2007	PWM dither value	OUT00…OUT11	0	ja	nach PreOp
0x2012	Input period duration	IN12…IN15	0	ja	nach PreOp
0x2013	Number of periods	IN12…IN15	0	ja	nach PreOp
0x2014	Period ratio value	IN12…IN15	0	ja	nach PreOp
0x2015	Input frequency	IN12…IN15	0.0	ja	nach PreOp
0x2016	Timebase frequency	IN12…IN15	50	ja	nach PreOp
0x2020	Input short to VBBS	IN00…IN11	0	ja	nach PreOp
0x2021	Input wire brake	IN00…IN11	0	ja	nach PreOp
0x2022	Output short circuit	OUT00…OUT07	0	ja	nach PreOp
0x2023	Output open circuit	OUT00…OUT07	0	ja	nach PreOp
0x2024	Output overload	OUT00…OUT01	0	ja	nach PreOp
0x2025	Input overcurrent	IN00…IN03	0	ja	nach PreOp
0x2030	Input resistance	IN04…IN05	0	ja	nach PreOp
0x2040	Supply voltage	VBBS	0	ja	nach PreOp
0x2041	Supply voltage	VBB1, VBB2	0	ja	nach PreOp
0x2050	Device temperature	device	0	ja	nach PreOp
0x20F0 != 0x20F1 *)	Node ID	device	124	sobald beide identisch	nach Reset
0x20F2 != 0x20F3 *)	Baud rate	device	3	sobald beide identisch	nach Reset
0x20F4	Autostart	device	0	ja	sofort
0x6000	Binary inputs	IN00…IN07 IN08…IN15	0	ja	nach PreOp
0x6200	Binary output	OUT00…OUT07 OUT08…OUT15	0	ja	nach PreOp
0x6404	Analog inputs	IN00…IN03	---	---	---
0x6414	Analog outputs	OUT00…OUT11	---	---	---

*) Werte müssen identisch sein!

Objektverzeichnis-Parametertabellen, Details

Inhalt

15982

Objektverzeichnis Pflichtobjekte (Index 0x1000…0x1FFF), Details

15985

Index	S-Idx	Parameter name	Data type		Default	Details
0x1000		Device type	ro	UDINT	0x000F 0191	Gerätetyp
0x1001		Error register	ro	USINT	0	Fehlerregister bitcodiert gemäß Profil 301 zulässige Werte: 0b0000 0000 = kein Fehler 0b0000 0001 = generic error 0b0001 0000 = communication error 0b1000 0000 = manufacturer specific
0x1018	0x0	Device identification Number of entries	ro	USINT	0x04	Geräteidentifizierung
	0x1	Vendor-ID	ro	UDINT	0x0690 7501	Vendor-ID des Geräts gemäß CiA-Spezifikation
	0x2	Product code	ro	STRING	0	Produkt-Code des Geräts
	0x3	Revision number	ro	UDINT	0	Revisionsnummer des Geräts
	0x4	Serial number	ro	UDINT	0	Seriennummer des Geräts

Legende:
Data type: ro = read only / rw = read and write / wo = write only

Objektverzeichnis optionale Objekte (Index 0x1000...0x10FF), Details

16603

Index	S-Idx	Parameter name	Data type		Default	Details
0x1003	0x0	Predefined error field Number of entries	rw	UDINT	0	Es wird eine Fehlerliste mit 4 Einträgen unterstützt
	0x1	Error history	ro	UDINT	0	Aufgetretener Fehler; codiert entsprechend EMCY-Liste der zuletzt aufgetretene Fehler steht jeweils in Sub-Index 1
	0x2	Error history	ro	UDINT	0	Aufgetretener Fehler; codiert entsprechend EMCY-Liste
	0x3	Error history	ro	UDINT	0	Aufgetretener Fehler; codiert entsprechend EMCY-Liste
	0x4	Error history	ro	UDINT	0	Aufgetretener Fehler; codiert entsprechend EMCY-Liste
	0x5	Error history	ro	UDINT	0	Aufgetretener Fehler; codiert entsprechend EMCY-Liste
0x1005		COB-ID synch message	rw	UDINT	0x0000 0080	Identifier der Synch Meldung Bit 30 = 0 ⇨ Gerät generiert keine Synch-Meldung Bit 30 = 1 ⇨ Gerät generiert eine Synch-Meldung Bit 29 = 0 ⇨ 11 Bit-ID Bit 29 = 1 ⇨ ID = 0x80 + Node-ID
0x1006		Communication cycle period	rw	UDINT	0	max. Zeit zwischen 2 Synch. Objekten in [µs] Nutzauflösung = 1 ms
0x1008		Manufacturer device name	ro	STRING	EXB01	Gerätebezeichnung
0x1009		Manufacturer hardware version	ro	STRING	V00.00.00	Hardware-Version
0x100A		Manufacturer software version	ro	STRING	V00.00.00	Software-Version
0x100C		Guard time	rw	UINT	0	Das Gerät erwartet innerhalb dieser Zeit in [ms] ein „node guarding" des Netz-Masters. 0 = diese Funktion wird nicht unterstützt. ! Die Überwachung des Knotens mit „node guarding" oder „heartbeat" ist nur alternativ möglich.
0x100D		Lifetime factor	rw	USINT	0	Falls für "guard time" • "lifetime" kein "node guarding" empfangen wurde, schaltet das Gerät die Ausgänge aus. Das Gerät wechselt den CANopen-Status nach PREOP. Vorgabe: "guard time" • "lifetime" = 0...65535
0x1010	0x0	Store parameters Largest sub-index supported	ro	USINT	0x01	Anzahl der Optionen "sichern"
	0x1	Save all parameters	rw	UDINT	2	Automatisches Sichern aller geänderten Parameter 0 = Autosicherung AUS 2 = Autosicherung EIN
0x1011	0x0	Restore default parameters Largest sub-index supported	ro	USINT	0x01	Anzahl der Optionen "Restore"
	0x1	Restore all default parameters	rw	UDINT	0x01	Wird hier der String "load" eingetragen, werden die Parameter mit den werkseitigen Voreinstellungen belegt und sind nach dem nächsten Reset gültig.
0x1014		COBId Emergency	rw	UDINT	0x80 + Node-ID	Bit 31 = 0 ⇨ EMCY ist gültig Bit 31 = 1 ⇨ EMCY ist nicht gültig Bit 29 = 0 ⇨ 11 Bit-ID Bit 29 = 1 ⇨ ID = 0x80 + Node-ID CAN-Identifier kann vom Benutzer geändert werden.
0x1016	0x0	Consumer heartbeat times Nums consumer heartbeat time	ro	USINT	0x01	Heartbeat-Überwachungszeit für den Knoten Anzahl der überwachten Geräte = 1

Index	S-Idx	Parameter name	Data type		Default	Details
	0x1	Consumer heartbeat time	rw	UDINT	0	Heartbeat-Überwachungszeit für den Knoten Format: 0x0nntttt tttt = Überwachungszeit [ms] nn = Knotennummer wenn nn=0 oder tttt=0 ⇨ keine Überwachung ❗ Die Überwachung des Knotens mit „node guarding" oder „heartbeat" ist nur alternativ möglich.
0x1017		Producer heartbeat time	rw	UINT	0	Zeitintervall [ms], in dem das Gerät einen Producer-Heartbeat erzeugt

Legende:
Data type: ro = read only / rw = read and write / wo = write only

Objektverzeichnis optionale Objekte (Index 0x1400...0x14FF), Details

16604

Receive PDO communication parameters

Index	S-Idx	Parameter name	Data type		Default	Details
0x1400	0x0	Receive PDO Communication Parameter Number of entries	ro	USINT	0x02	Receive PDO 1: Binärausgänge Anzahl der Einträge = 2
	0x1	COBID used by PDO	rw	UDINT	0x200 + Node-ID	CAN-ID des 1. Lese-PDOs Bit 31 = 0 ⇨ PDO ist gültig Bit 31 = 1 ⇨ PDO ist ungültig
	0x2	transmission type	rw	USINT	0x01	0x00 = synch acyclic 0x01...0xF0 = synch cyclic; Ausgänge werden erst nach „n" Synch Objekten aktualisiert n = 1...240 = 0x01...0xF0 0xFC/0xFD nicht implementiert 0xFE = asynch man. spec. event; Ausgänge werden sofort aktualisiert 0xFF = asynch device profile event; Ausgänge werden sofort aktualisiert
0x1401	0x0	Receive PDO Communication Parameter Number of entries	ro	USINT	0x02	Receive PDO 2: PWM-Ausgänge Anzahl der Einträge = 2
	0x1	COBID used by PDO	rw	UDINT	0x300 + Node-ID	CAN-ID des 2. Lese-PDOs Bit 31 = 0 ⇨ PDO ist gültig Bit 31 = 1 ⇨ PDO ist ungültig
	0x2	transmission type	rw	USINT	0x01	0x00 = synch acyclic 0x01...0xF0 = synch cyclic; Ausgänge werden erst nach „n" Synch Objekten aktualisiert n = 1...240 = 0x01...0xF0 0xFC/0xFD nicht implementiert 0xFE = asynch man. spec. event; Ausgänge werden sofort aktualisiert 0xFF = asynch device profile event; Ausgänge werden sofort aktualisiert
0x1402	0x0	Receive PDO Communication Parameter Number of entries	ro	USINT	0x02	Receive PDO 3: PWM-Ausgänge Anzahl der Einträge = 2
	0x1	COBID used by PDO	rw	UDINT	0x400 + Node-ID	CAN-ID des 3. Lese-PDOs Bit 31 = 0 ⇨ PDO ist gültig Bit 31 = 1 ⇨ PDO ist ungültig
	0x2	transmission type	rw	USINT	0x01	0x00 = synch acyclic 0x01...0xF0 = synch cyclic; Ausgänge werden erst nach „n" Synch Objekten aktualisiert n = 1...240 = 0x01...0xF0 0xFC/0xFD nicht implementiert 0xFE = asynch man. spec. event; Ausgänge werden sofort aktualisiert 0xFF = asynch device profile event; Ausgänge werden sofort aktualisiert
0x1403	0x0	Receive PDO Communication Parameter Number of entries	ro	USINT	0x02	Receive PDO 4: PWM-Ausgänge Anzahl der Einträge = 2
	0x1	COBID used by PDO	rw	UDINT	0x500 + Node-ID	CAN-ID des 4. Lese-PDOs Bit 31 = 0 ⇨ PDO ist gültig Bit 31 = 1 ⇨ PDO ist ungültig

Index	S-Idx	Parameter name	Data type		Default	Details
	0x2	transmission type	rw	USINT	0x01	0x00 = synch acyclic 0x01...0xF0 = synch cyclic; Ausgänge werden erst nach „n" Synch Objekten aktualisiert n = 1…240 = 0x01...0xF0 0xFC/0xFD nicht implementiert 0xFE = asynch man. spec. event; Ausgänge werden sofort aktualisiert 0xFF = asynch device profile event; Ausgänge werden sofort aktualisiert

Legende:
Data type: ro = read only / rw = read and write / wo = write only

Objektverzeichnis optionale Objekte (Index 0x1600...0x16FF), Details

16605

Receive PDO mapping

Index	S-Idx	Parameter name	Data type		Default	Details
0x1600	0x0	Receive PDO mapping Number of mapped objects in PDO	rw	USINT	0x02	Mapping Lese-PDO 1: Binärausgänge Anzahl der eingebundenen Applikationsobjekte = 2
	0x1	PDO mapping	ro	UDINT	0x6200 0108	im Index 0x6200, SubIndex 01 steht 1 Byte Binär-Eingänge IN00...IN07 0b---- ---X = IN00 0b---- --X- = IN01 0b---- -X-- = IN02 0b---- X--- = IN03 0b---X ---- = IN04 0b--X- ---- = IN05 0b-X-- ---- = IN06 0bX--- ---- = IN07
	0x2	PDO mapping	ro	UDINT	0x6200 0208	im Index 0x6200, SubIndex 02 steht 1 Byte Binär-Eingänge IN08...IN15 0b---- ---X = IN08 0b---- --X- = IN09 0b---- -X-- = IN10 0b---- X--- = IN11 0b---X ---- = IN12 0b--X- ---- = IN13 0b-X-- ---- = IN14 0bX--- ---- = IN15
0x1601	0x0	Receive PDO mapping Number of mapped objects in PDO	rw	USINT	0x04	Mapping Lese-PDO 2: PWM-Ausgänge OUT00...OUT03 Anzahl der eingebundenen Applikationsobjekte = 4
	0x1	PDO mapping	rw	UDINT	0x6414 0110	PWM-Ausgang OUT00 Im Index 0x6414, SubIndex 0x1 steht der Sollwert des PWM-Ausgangs OUT00, der Wert wird als Tastverhältnis in ‰ oder als Stromsollwert interpretiert (abhängig von Konfiguration Index 0x2000).
	0x2	PDO mapping	rw	UDINT	0x6414 0210	PWM-Ausgang OUT01 Im Index 0x6414, SubIndex 0x2 steht der Sollwert des PWM-Ausgangs OUT01, der Wert wird als Tastverhältnis in ‰ oder als Stromsollwert interpretiert (abhängig von Konfiguration Index 0x2000).
	0x3	PDO mapping	rw	UDINT	0x6414 0310	PWM-Ausgang OUT02 Im Index 0x6414, SubIndex 0x3 steht der Sollwert des PWM-Ausgangs OUT02, der Wert wird als Tastverhältnis in ‰ oder als Stromsollwert interpretiert (abhängig von Konfiguration Index 0x2000).
	0x4	PDO mapping	rw	UDINT	0x6414 0410	PWM-Ausgang OUT03 Im Index 0x6414, SubIndex 0x4 steht der Sollwert des PWM-Ausgangs OUT03, der Wert wird als Tastverhältnis in ‰ oder als Stromsollwert interpretiert (abhängig von Konfiguration Index 0x2000).
0x1602	0x0	Receive PDO mapping Number of mapped objects in PDO	rw	USINT	0x04	Mapping Lese-PDO 3: PWM-Ausgänge OUT04...OUT07 Anzahl der eingebundenen Applikationsobjekte = 4
	0x1	PDO mapping	rw	UDINT	0x6414 0510	PWM-Ausgang OUT04 Im Index 0x6414, SubIndex 0x5 steht der Sollwert des PWM-Ausgangs OUT04, der Wert wird als Tastverhältnis in ‰ oder als Stromsollwert interpretiert (abhängig von Konfiguration Index 0x2000).

Index	S-Idx	Parameter name	Data type		Default	Details
	0x2	PDO mapping	rw	UDINT	0x6414 0610	PWM-Ausgang OUT05 Im Index 0x6414, SubIndex 0x6 steht der Sollwert des PWM-Ausgangs OUT05, der Wert wird als Tastverhältnis in ‰ oder als Stromsollwert interpretiert (abhängig von Konfiguration Index 0x2000).
	0x3	PDO mapping	rw	UDINT	0x6414 0710	PWM-Ausgang OUT06 Im Index 0x6414, SubIndex 0x7 steht der Sollwert des PWM-Ausgangs OUT06, der Wert wird als Tastverhältnis in ‰ oder als Stromsollwert interpretiert (abhängig von Konfiguration Index 0x2000).
	0x4	PDO mapping	rw	UDINT	0x6414 0810	PWM-Ausgang OUT07 Im Index 0x6414, SubIndex 0x8 steht der Sollwert des PWM-Ausgangs OUT07, der Wert wird als Tastverhältnis in ‰ oder als Stromsollwert interpretiert (abhängig von Konfiguration Index 0x2000).
	0x5	PDO mapping	rw	UDINT	0	Reserve
0x1603	0x0	Receive PDO mapping Number of mapped objects in PDO	rw	USINT	0x04	Mapping Lese-PDO 4: PWM-Ausgänge OUT08…OUT11 Anzahl der eingebundenen Applikationsobjekte = 4
	0x1	PDO mapping	rw	UDINT	0x6414 0910	PWM-Ausgang OUT08 Im Index 0x6414, SubIndex 0x9 steht der Sollwert des PWM-Ausgangs OUT08, der Wert wird als Tastverhältnis in ‰ oder als Stromsollwert interpretiert (abhängig von Konfiguration Index 0x2000).
	0x2	PDO mapping	rw	UDINT	0x6414 0A10	PWM-Ausgang OUT09 Im Index 0x6414, SubIndex 0xA steht der Sollwert des PWM-Ausgangs OUT09, der Wert wird als Tastverhältnis in ‰ oder als Stromsollwert interpretiert (abhängig von Konfiguration Index 0x2000).
	0x3	PDO mapping	rw	UDINT	0x6414 0B10	PWM-Ausgang OUT10 Im Index 0x6414, SubIndex 0xB steht der Sollwert des PWM-Ausgangs OUT10, der Wert wird als Tastverhältnis in ‰ oder als Stromsollwert interpretiert (abhängig von Konfiguration Index 0x2000).
	0x4	PDO mapping	rw	UDINT	0x6414 0C10	PWM-Ausgang OUT11 Im Index 0x6414, SubIndex 0xC steht der Sollwert des PWM-Ausgangs OUT11, der Wert wird als Tastverhältnis in ‰ oder als Stromsollwert interpretiert (abhängig von Konfiguration Index 0x2000).
	0x5	PDO mapping	rw	UDINT	0	Reserve

Legende:
Data type: ro = read only / rw = read and write / wo = write only

Objektverzeichnis optionale Objekte (Index 0x1800…0x18FF), Details

16606

Transmit PDO communication parameters

Index	S-Idx	Parameter name	Data type		Default	Details
0x1800	0x0	Transmit PDO Communication Parameter Number of entries	ro	USINT	0x05	Konfiguration Sende-PDO 1 Anzahl der Einträge = 5
	0x1	COBID used by PDO	rw	UDINT	0x180 + Node-ID	CAN-ID des Sende-PDO 1 Bit 31 = 0 ⇨ PDO ist gültig Bit 31 = 1 ⇨ PDO ist ungültig
	0x2	transmission type	rw	USINT	0x01	0x00 = synch acyclic 0x01...0xF0 = synch cyclic; Werte werden erst nach „n" Synch Objekten übertragen n = 1…240 = 0x01...0xF0 0xFC/0xFD nicht implementiert 0xFE = asynch man. spec. event; Werte werden sofort übertragen 0xFF = asynch device profile event; Werte werden sofort übertragen
	0x3	inhibit time	rw	UINT	0	Wartezeit im Sende-Type "asynch" bevor das PDO frühestens wieder gesendet wird. (0...65535 • 100 µs)
	0x4	reserved	rw	USINT	0	Reserve
	0x5	event time	rw	UINT	0	max. Sendepause im Sende-Type „asynch" (0...65535 ms) Nach Ablauf dieser Zeit wird das PDO übertragen, auch wenn das Appl.-Event nicht eingetreten ist.
0x1801	0x0	Transmit PDO Communication Parameter Number of entries	ro	USINT	0x05	Konfiguration Sende-PDO 2 Anzahl der Einträge = 5
	0x1	COBID used by PDO	rw	UDINT	0x280 + Node-ID	CAN-ID des Sende-PDO 2 Bit 31 = 0 ⇨ PDO ist gültig Bit 31 = 1 ⇨ PDO ist ungültig
	0x2	transmission type	rw	USINT	0x01	0x00 = synch acyclic 0x01...0xF0 = synch cyclic; Werte werden erst nach „n" Synch Objekten übertragen n = 1…240 = 0x01...0xF0 0xFC/0xFD nicht implementiert 0xFE = asynch man. spec. event; Werte werden sofort übertragen 0xFF = asynch device profile event; Werte werden sofort übertragen
	0x3	inhibit time	rw	UINT	0	Wartezeit im Sende-Type "asynch" bevor das PDO frühestens wieder gesendet wird. (0...65535 • 100 µs)
	0x4	reserved	rw	USINT	0	Reserve
	0x5	event time	rw	UINT	0	max. Sendepause im Sende-Type „asynch" (0...65535 ms) Nach Ablauf dieser Zeit wird das PDO übertragen, auch wenn das Appl.-Event nicht eingetreten ist.
0x1802	0x0	Transmit PDO Communication Parameter Number of entries	ro	USINT	0x05	Konfiguration Sende-PDO 3 Anzahl der Einträge = 5
	0x1	COBID used by PDO	rw	UDINT	0x380 + Node-ID	CAN-ID des Sende-PDO 3 Bit 31 = 0 ⇨ PDO ist gültig Bit 31 = 1 ⇨ PDO ist ungültig

Index	S-Idx	Parameter name	Data type		Default	Details
	0x2	transmission type	rw	USINT	0x01	0x00 = synch acyclic 0x01...0xF0 = synch cyclic; Werte werden erst nach „n" Synch Objekten übertragen n = 1…240 = 0x01...0xF0 0xFC/0xFD nicht implementiert 0xFE = asynch man. spec. event; Werte werden sofort übertragen 0xFF = asynch device profile event; Werte werden sofort übertragen
	0x3	inhibit time	rw	UINT	0	Wartezeit im Sende-Type "asynch" bevor das PDO frühestens wieder gesendet wird. (0...65535 • 100 µs)
	0x4	reserved	rw	USINT	0	Reserve
	0x5	event time	rw	UINT	0	max. Sendepause im Sende-Type „asynch" (0...65535 ms) Nach Ablauf dieser Zeit wird das PDO übertragen, auch wenn das Appl.-Event nicht eingetreten ist.
0x1803	0x0	Transmit PDO Communication Parameter Number of entries	ro	USINT	0x05	Konfiguration Sende-PDO 4 Anzahl der Einträge = 5
	0x1	COBID used by PDO	rw	UDINT	0x480 + Node-ID	CAN-ID des Sende-PDO 4 Bit 31 = 0 ⇨ PDO ist gültig Bit 31 = 1 ⇨ PDO ist ungültig
	0x2	transmission type	rw	USINT	0x01	0x00 = synch acyclic 0x01...0xF0 = synch cyclic; Werte werden erst nach „n" Synch Objekten übertragen n = 1…240 = 0x01...0xF0 0xFC/0xFD nicht implementiert 0xFE = asynch man. spec. event; Werte werden sofort übertragen 0xFF = asynch device profile event; Werte werden sofort übertragen
	0x3	inhibit time	rw	UINT	0	Wartezeit im Sende-Type "asynch" bevor das PDO frühestens wieder gesendet wird. (0...65535 • 100 µs)
	0x4	reserved	rw	USINT	0	Reserve
	0x5	event time	rw	UINT	0	max. Sendepause im Sende-Type „asynch" (0...65535 ms) Nach Ablauf dieser Zeit wird das PDO übertragen, auch wenn das Appl.-Event nicht eingetreten ist.
0x1804	0x0	Transmit PDO Communication Parameter Number of entries	ro	USINT	0x05	Konfiguration Sende-PDO 5 Anzahl der Einträge = 5
	0x1	COBID used by PDO	rw	UDINT	0x181 + Node-ID	CAN-ID des Sende-PDO 5 Bit 31 = 0 ⇨ PDO ist gültig Bit 31 = 1 ⇨ PDO ist ungültig
	0x2	transmission type	rw	USINT	0x01	0x00 = synch acyclic 0x01...0xF0 = synch cyclic; Werte werden erst nach „n" Synch Objekten übertragen n = 1…240 = 0x01...0xF0 0xFC/0xFD nicht implementiert 0xFE = asynch man. spec. event; Werte werden sofort übertragen 0xFF = asynch device profile event; Werte werden sofort übertragen
	0x3	inhibit time	rw	UINT	0	Wartezeit im Sende-Type "asynch" bevor das PDO frühestens wieder gesendet wird. (0...65535 • 100 µs)
	0x4	reserved	rw	USINT	0	Reserve

Index	S-Idx	Parameter name	Data type		Default	Details
	0x5	event time	rw	UINT	0	max. Sendepause im Sende-Type „asynch" (0...65535 ms) Nach Ablauf dieser Zeit wird das PDO übertragen, auch wenn das Appl.-Event nicht eingetreten ist.
0x1805	0x0	Transmit PDO Communication Parameter Number of entries	ro	USINT	0x05	Konfiguration Sende-PDO 6 Anzahl der Einträge = 5
	0x1	COBID used by PDO	rw	UDINT	0x281 + Node-ID	CAN-ID des Sende-PDO 6 Bit 31 = 0 ⇨ PDO ist gültig Bit 31 = 1 ⇨ PDO ist ungültig
	0x2	transmission type	rw	USINT	0x01	0x00 = synch acyclic 0x01...0xF0 = synch cyclic; Werte werden erst nach „n" Synch Objekten übertragen n = 1…240 = 0x01...0xF0 0xFC/0xFD nicht implementiert 0xFE = asynch man. spec. event; Werte werden sofort übertragen 0xFF = asynch device profile event; Werte werden sofort übertragen
	0x3	inhibit time	rw	UINT	0	Wartezeit im Sende-Type "asynch" bevor das PDO frühestens wieder gesendet wird. (0...65535 • 100 µs)
	0x4	reserved	rw	USINT	0	Reserve
	0x5	event time	rw	UINT	0	max. Sendepause im Sende-Type „asynch" (0...65535 ms) Nach Ablauf dieser Zeit wird das PDO übertragen, auch wenn das Appl.-Event nicht eingetreten ist.
0x1806	0x0	Transmit PDO Communication Parameter Number of entries	ro	USINT	0x05	Konfiguration Sende-PDO 7 Anzahl der Einträge = 5
	0x1	COBID used by PDO	rw	UDINT	0x381 + Node-ID	CAN-ID des Sende-PDO 7 Bit 31 = 0 ⇨ PDO ist gültig Bit 31 = 1 ⇨ PDO ist ungültig
	0x2	transmission type	rw	USINT	0x01	0x00 = synch acyclic 0x01...0xF0 = synch cyclic; Werte werden erst nach „n" Synch Objekten übertragen n = 1…240 = 0x01...0xF0 0xFC/0xFD nicht implementiert 0xFE = asynch man. spec. event; Werte werden sofort übertragen 0xFF = asynch device profile event; Werte werden sofort übertragen
	0x3	inhibit time	rw	UINT	0	Wartezeit im Sende-Type "asynch" bevor das PDO frühestens wieder gesendet wird. (0...65535 • 100 µs)
	0x4	reserved	rw	USINT	0	Reserve
	0x5	event time	rw	UINT	0	max. Sendepause im Sende-Type „asynch" (0...65535 ms) Nach Ablauf dieser Zeit wird das PDO übertragen, auch wenn das Appl.-Event nicht eingetreten ist.
0x1807	0x0	Transmit PDO Communication Parameter Number of entries	ro	USINT	0x05	Konfiguration Sende-PDO 8 Anzahl der Einträge = 5
	0x1	COBID used by PDO	rw	UDINT	0x481 + Node-ID	CAN-ID des Sende-PDO 8 Bit 31 = 0 ⇨ PDO ist gültig Bit 31 = 1 ⇨ PDO ist ungültig

Index	S-Idx	Parameter name	Data type		Default	Details
	0x2	transmission type	rw	USINT	0x01	0x00 = synch acyclic 0x01...0xF0 = synch cyclic; Werte werden erst nach „n" Synch Objekten übertragen n = 1…240 = 0x01...0xF0 0xFC/0xFD nicht implementiert 0xFE = asynch man. spec. event; Werte werden sofort übertragen 0xFF = asynch device profile event; Werte werden sofort übertragen
	0x3	inhibit time	rw	UINT	0	Wartezeit im Sende-Type "asynch" bevor das PDO frühestens wieder gesendet wird. (0...65535 • 100 µs)
	0x4	reserved	rw	USINT	0	Reserve
	0x5	event time	rw	UINT	0	max. Sendepause im Sende-Type „asynch" (0...65535 ms) Nach Ablauf dieser Zeit wird das PDO übertragen, auch wenn das Appl.-Event nicht eingetreten ist.
0x1808	0x0	Transmit PDO Communication Parameter Number of entries	ro	USINT	0x05	Konfiguration Sende-PDO 9 Anzahl der Einträge = 5
	0x1	COBID used by PDO	rw	UDINT	0x181 + Node-ID	CAN-ID des Sende-PDO 9 Bit 31 = 0 ⇨ PDO ist gültig Bit 31 = 1 ⇨ PDO ist ungültig
	0x2	transmission type	rw	USINT	0x01	0x00 = synch acyclic 0x01...0xF0 = synch cyclic; Werte werden erst nach „n" Synch Objekten übertragen n = 1…240 = 0x01...0xF0 0xFC/0xFD nicht implementiert 0xFE = asynch man. spec. event; Werte werden sofort übertragen 0xFF = asynch device profile event; Werte werden sofort übertragen
	0x3	inhibit time	rw	UINT	0	Wartezeit im Sende-Type "asynch" bevor das PDO frühestens wieder gesendet wird. (0...65535 • 100 µs)
	0x4	reserved	rw	USINT	0	Reserve
	0x5	event time	rw	UINT	0	max. Sendepause im Sende-Type „asynch" (0...65535 ms) Nach Ablauf dieser Zeit wird das PDO übertragen, auch wenn das Appl.-Event nicht eingetreten ist.
0x1809	0x0	Transmit PDO Communication Parameter Number of entries	ro	USINT	0x05	Konfiguration Sende-PDO 10 Anzahl der Einträge = 5
	0x1	COBID used by PDO	rw	UDINT	0x281 + Node-ID	CAN-ID des Sende-PDO 10 Bit 31 = 0 ⇨ PDO ist gültig Bit 31 = 1 ⇨ PDO ist ungültig
	0x2	transmission type	rw	USINT	0x01	0x00 = synch acyclic 0x01...0xF0 = synch cyclic; Werte werden erst nach „n" Synch Objekten übertragen n = 1…240 = 0x01...0xF0 0xFC/0xFD nicht implementiert 0xFE = asynch man. spec. event; Werte werden sofort übertragen 0xFF = asynch device profile event; Werte werden sofort übertragen
	0x3	inhibit time	rw	UINT	0	Wartezeit im Sende-Type "asynch" bevor das PDO frühestens wieder gesendet wird. (0...65535 • 100 µs)
	0x4	reserved	rw	USINT	0	Reserve

Index	S-Idx	Parameter name	Data type		Default	Details
	0x5	event time	rw	UINT	0	max. Sendepause im Sende-Type „asynch" (0...65535 ms) Nach Ablauf dieser Zeit wird das PDO übertragen, auch wenn das Appl.-Event nicht eingetreten ist.

Legende:
Data type: ro = read only / rw = read and write / wo = write only

Objektverzeichnis optionale Objekte (Index 0x1A00...0x1AFF), Details

16607

Transmit PDO mapping

Index	S-Idx	Parameter name	Data type		Default	Details
0x1A00	0x0	Transmit PDO mapping Number of mapped objects in PDO	rw	USINT	0x07	Mapping Sende-PDO 1 Anzahl der eingebundenen Applikationsobjekte = 7
	0x1	PDO mapping	rw	UDINT	0x6000 0108	Index 0x6000, SubIndex 0x1 Binär-Eingänge 00...07: Istwerte (Bit-codiert)
	0x2	PDO mapping	rw	UDINT	0x6000 0208	Index 0x6000, SubIndex 0x2 Binär-Eingänge 08...15: Istwerte (Bit-codiert)
	0x3	PDO mapping	rw	UDINT	0x2020 0108	Index 0x2020, SubIndex 0x1 Eingänge 00...07: Merker "Kurzschluss" (Bit-codiert)
	0x4	PDO mapping	rw	UDINT	0x2020 0208	Index 0x2020, SubIndex 0x2 Eingänge 08...11: Merker "Kurzschluss" (Bit-codiert)
	0x5	PDO mapping	rw	UDINT	0x2021 0108	Index 0x2021, SubIndex 0x1 Eingänge 00...07: Merker "Leiterbruch" (Bit-codiert)
	0x6	PDO mapping	rw	UDINT	0x2021 0208	Index 0x2021, SubIndex 0x2 Eingänge 08...11: Merker "Leiterbruch" (Bit-codiert)
	0x7	PDO mapping	rw	UDINT	0x2025 0108	Index 0x2025, SubIndex 0x1 Eingänge 00...03: Merker "Überlast" (Bit-codiert)
0x1A01	0x0	Transmit PDO mapping Number of mapped objects in PDO	rw	USINT	0x04	Mapping Sende-PDO 2 (Analog-Eingänge) Anzahl der eingebundenen Applikationsobjekte = 4
	0x1	PDO mapping	rw	UDINT	0x6404 0110	Index 0x6404, SubIndex 0x1 Analog-Eingang 00: Istwert (abhängig von Konfiguration 0x2000)
	0x2	PDO mapping	rw	UDINT	0x6404 0210	Index 0x6404, SubIndex 0x2 Analog-Eingang 01: Istwert (abhängig von Konfiguration 0x2000)
	0x3	PDO mapping	rw	UDINT	0x6404 0310	Index 0x6404, SubIndex 0x3 Analog-Eingang 02: Istwert (abhängig von Konfiguration 0x2000)
	0x4	PDO mapping	rw	UDINT	0x6404 0410	Index 0x6404, SubIndex 0x4 Analog-Eingang 03: Istwert (abhängig von Konfiguration 0x2000)
0x1A02	0x0	Transmit PDO mapping Number of mapped objects in PDO	rw	USINT	0x04	Mapping Sende-PDO 3 Anzahl der eingebundenen Applikationsobjekte = 4
	0x1	PDO mapping	rw	UDINT	0x2030 0110	Index 0x2030, SubIndex 0x1 Eingang 04: Widerstands-Istwert
	0x2	PDO mapping	rw	UDINT	0x2030 0210	Index 0x2030, SubIndex 0x2 Eingang 05: Widerstands-Istwert
	0x3	PDO mapping	rw	UDINT	0x2002 0110	Index 0x2002, SubIndex 0x1 Ausgang 00: Strom-Istwert
	0x4	PDO mapping	rw	UDINT	0x2002 0210	Index 0x2002, SubIndex 0x2 Ausgang 01: Strom-Istwert
	0x5	PDO mapping	rw	UDINT	0	Reserve
0x1A03	0x0	Transmit PDO mapping Number of mapped objects in PDO	rw	USINT	0x02	Mapping Sende-PDO 4 (Periodendauer IN12...IN13) Anzahl der eingebundenen Applikationsobjekte = 2
	0x1	PDO mapping	rw	UDINT	0x2012 0120	Index 0x2012, SubIndex 0x1 Frequenzeingang 12: Periodendauer des Signals
	0x2	PDO mapping	rw	UDINT	0x2012 0220	Index 0x2012, SubIndex 0x2 Frequenzeingang 13: Periodendauer des Signals

Index	S-Idx	Parameter name	Data type		Default	Details
	0x3	PDO mapping	rw	UDINT	0	Reserve
0x1A04	0x0	Transmit PDO mapping Number of mapped objects in PDO	rw	USINT	0x02	Mapping Sende-PDO 5 (Periodendauer IN14...IN15) Anzahl der eingebundenen Applikationsobjekte = 2
	0x1	PDO mapping	rw	UDINT	0x2012 0320	Index 0x2012, SubIndex 0x3 Frequenzeingang 14: Periodendauer des Signals
	0x2	PDO mapping	rw	UDINT	0x2012 0420	Index 0x2012, SubIndex 0x4 Frequenzeingang 15: Periodendauer des Signals
	0x3	PDO mapping	rw	UDINT	0	Reserve
0x1A05	0x0	Transmit PDO mapping Number of mapped objects in PDO	rw	USINT	0x04	Mapping Sende-PDO 6 (Einschaltzeit des Signals an Frequenzeingang IN12...IN15) Anzahl der eingebundenen Applikationsobjekte = 4
	0x1	PDO mapping	rw	UDINT	0x2014 0110	Index 0x2014, SubIndex 0x1 Frequenzeingang 12: Einschaltzeit des Signals in ‰
	0x2	PDO mapping	rw	UDINT	0x2014 0210	Index 0x2014, SubIndex 0x2 Frequenzeingang 13: Einschaltzeit des Signals in ‰
	0x3	PDO mapping	rw	UDINT	0x2014 0310	Index 0x2014, SubIndex 0x3 Frequenzeingang 14: Einschaltzeit des Signals in ‰
	0x4	PDO mapping	rw	UDINT	0x2014 0410	Index 0x2014, SubIndex 0x4 Frequenzeingang 15: Einschaltzeit des Signals in ‰
	0x5	PDO mapping	rw	UDINT	0	Reserve
0x1A06	0x0	Transmit PDO mapping Number of mapped objects in PDO	rw	USINT	0x02	Mapping Sende-PDO 7 (Frequenz an IN12...IN13) Anzahl der eingebundenen Applikationsobjekte = 2
	0x1	PDO mapping	rw	UDINT	0x2015 0120	Index 0x2015, SubIndex 0x1 Frequenzeingang 12: Frequenzwert des Signals in Hz
	0x2	PDO mapping	rw	UDINT	0x2015 0220	Index 0x2015, SubIndex 0x2 Frequenzeingang 13: Frequenzwert des Signals in Hz
	0x3	PDO mapping	rw	UDINT	0	Reserve
0x1A07	0x0	Transmit PDO mapping Number of mapped objects in PDO	rw	USINT	0x02	Mapping Sende-PDO 8 (Frequenz an IN14...IN15) Anzahl der eingebundenen Applikationsobjekte = 2
	0x1	PDO mapping	rw	UDINT	0x2015 0320	Index 0x2015, SubIndex 0x3 Frequenzeingang 14: Frequenzwert des Signals in Hz
	0x2	PDO mapping	rw	UDINT	0x2015 0420	Index 0x2015, SubIndex 0x4 Frequenzeingang 15: Frequenzwert des Signals in Hz
	0x3	PDO mapping	rw	UDINT	0	Reserve
0x1A08	0x0	Transmit PDO mapping Number of mapped objects in PDO	rw	USINT	0x03	Mapping Sende-PDO 9 (Fehlermerker OUT00...OUT07) Anzahl der eingebundenen Applikationsobjekte = 3
	0x1	PDO mapping	rw	UDINT	0x2022 0108	Index 0x2022, SubIndex 0x1 OUT00...OUT07: Merker "Kurzschluss" (Bit-codiert)
	0x2	PDO mapping	rw	UDINT	0x2023 0108	Index 0x2023, SubIndex 0x1 OUT00...OUT07: Merker "Leiterbruch" (Bit-codiert)
	0x3	PDO mapping	rw	UDINT	0x2024 0108	Index 0x2024, SubIndex 0x1 OUT00...OUT01: Merker "Überlast" (Bit-codiert)
	0x4	PDO mapping	rw	UDINT	0	Reserve
0x1A09	0x0	Transmit PDO mapping Number of mapped objects in PDO	rw	USINT	0x04	Mapping Sende-PDO 10 (Systemmerker) Anzahl der eingebundenen Applikationsobjekte = 4
	0x1	PDO mapping	rw	UDINT	0x2040 0110	Index 0x2040, SubIndex 0x1 System-Versorgungsspannung VBBS

Index	S-Idx	Parameter name	Data type		Default	Details
	0x2	PDO mapping	rw	UDINT	0x2041 0110	Index 0x2041, SubIndex 0x1 Ausgangs-Versorgungsspannung VBB1
	0x3	PDO mapping	rw	UDINT	0x2041 0210	Index 0x2041, SubIndex 0x2 Ausgangs-Versorgungsspannung VBB2
	0x4	PDO mapping	rw	UDINT	0x2050 0010	Index 0x2050, SubIndex 0x0 Systemtemperatur in °C
	0x5	PDO mapping	rw	UDINT	0	Reserve

Legende:
Data type: ro = read only / rw = read and write / wo = write only

Objektverzeichnis herstellerspezifische Objekte (Index 0x2000...0x6FFF), Details

15983

Index	S-Idx	Parameter name	Data type		Default	Details	
0x2000	0x0	IO configuration Largest sub-index supported	ro	USINT	32	Konfiguration Ein- / Ausgänge Größter unterstützter Sub-Index = 32	
	0x1	Configuration IN00	rw	USINT	10	0 = 0x00 3 = 0x03 6 = 0x06 7 = 0x07 9 = 0x09 10 = 0x0A 11 = 0x0B 12 = 0x0C	off Input IN00 0...10 000 mV ratiometric 0...1000 ‰ 0...20 000 µA 0...32 000 mV binary plus switched binary plus switched with diagnosis binary minus switched
	0x2	Configuration IN01	rw	USINT	10	0 = 0x00 3 = 0x03 6 = 0x06 7 = 0x07 9 = 0x09 10 = 0x0A 11 = 0x0B 12 = 0x0C	off Input IN01 0...10 000 mV ratiometric 0...1000 ‰ 0...20 000 µA 0...32 000 mV binary plus switched binary plus switched with diagnosis binary minus switched
	0x3	Configuration IN02	rw	USINT	10	0 = 0x00 3 = 0x03 6 = 0x06 7 = 0x07 9 = 0x09 10 = 0x0A 11 = 0x0B 12 = 0x0C	off Input IN02 0...10 000 mV ratiometric 0...1000 ‰ 0...20 000 µA 0...32 000 mV binary plus switched binary plus switched with diagnosis binary minus switched
	0x4	Configuration IN03	rw	USINT	10	0 = 0x00 3 = 0x03 6 = 0x06 7 = 0x07 9 = 0x09 10 = 0x0A 11 = 0x0B 12 = 0x0C	off Input IN03 0...10 000 mV ratiometric 0...1000 ‰ 0...20 000 µA 0...32 000 mV binary plus switched binary plus switched with diagnosis binary minus switched
0x2000	0x5	Configuration IN04	rw	USINT	10	0 = 0x00 10 = 0x0A 11 = 0x0B 18 = 0x12	off Input IN04 binary plus switched binary plus switched with diagnosis 16...30 000 Ohm
	0x6	Configuration IN05	rw	USINT	10	0 = 0x00 10 = 0x0A 11 = 0x0B 18 = 0x12	off Input IN05 binary plus switched binary plus switched with diagnosis 16...30 000 Ohm
0x2000	0x7	Configuration IN06	rw	USINT	10	0 = 0x00 10 = 0x0A 11 = 0x0B	off Input IN06 binary plus switched binary plus switched with diagnosis
	0x8	Configuration IN07	rw	USINT	10	0 = 0x00 10 = 0x0A 11 = 0x0B	off Input IN07 binary plus switched binary plus switched with diagnosis
	0x9	Configuration IN08	rw	USINT	10	0 = 0x00 10 = 0x0A 11 = 0x0B	off Input IN08 binary plus switched binary plus switched with diagnosis
	0xA	Configuration IN09	rw	USINT	10	0 = 0x00 10 = 0x0A 11 = 0x0B	off Input IN09 binary plus switched binary plus switched with diagnosis
	0xB	Configuration IN10	rw	USINT	10	0 = 0x00 10 = 0x0A 11 = 0x0B	off Input IN10 binary plus switched binary plus switched with diagnosis

Index	S-Idx	Parameter name		Data type	Default	Details	
	0xC	Configuration IN11	rw	USINT	10	0 = 0x00 10 = 0x0A 11 = 0x0B	off Input IN11 binary plus switched binary plus switched with diagnosis
0x2000	0xD	Configuration IN12	rw	USINT	1	0 = 0x00 1 = 0x01 14 = 0x0E 19 = 0x13 20 = 0x14 21 = 0x15 22 = 0x16 23 = 0x17	off Input IN12 binary plus switched, digitally monitored frequency 0…30 000 Hz period duration period duration as ratio 0…1 000 ‰ counting up counting down incremental encoder
	0xE	Configuration IN13	rw	USINT	1	0 = 0x00 1 = 0x01 14 = 0x0E 19 = 0x13 20 = 0x14 21 = 0x15 22 = 0x16 23 = 0x17	off Input IN13 binary plus switched, digitally monitored frequency 0…30 000 Hz period duration period duration as ratio 0…1 000 ‰ counting up counting down incremental encoder
	0xF	Configuration IN14	rw	USINT	1	0 = 0x00 1 = 0x01 14 = 0x0E 19 = 0x13 20 = 0x14 21 = 0x15 22 = 0x16 23 = 0x17	off Input IN14 binary plus switched, digitally monitored frequency 0…30 000 Hz period duration period duration as ratio 0…1 000 ‰ counting up counting down incremental encoder
	0x10	Configuration IN15	rw	USINT	1	0 = 0x00 1 = 0x01 14 = 0x0E 19 = 0x13 20 = 0x14 21 = 0x15 22 = 0x16 23 = 0x17	off Input IN15 binary plus switched, digitally monitored frequency 0…30 000 Hz period duration period duration as ratio 0…1 000 ‰ counting up counting down incremental encoder
0x2000	0x11	Configuration OUT00	rw	USINT	2	0 = 0x00 2 = 0x02 4 = 0x04 5 = 0x05 15 = 0x0F 16 = 0x10	off Input OUT00 binary plus switched PWM output current control binary plus switched with diagnosis binary plus switched with diagnosis + protection
	0x12	Configuration OUT01	rw	USINT	2	0 = 0x00 2 = 0x02 4 = 0x04 5 = 0x05 15 = 0x0F 16 = 0x10	off Input OUT01 binary plus switched PWM output current control binary plus switched with diagnosis binary plus switched with diagnosis + protection
0x2000	0x13	Configuration OUT02	rw	USINT	2	0 = 0x00 2 = 0x02 4 = 0x04 15 = 0x0F 16 = 0x10	off Input OUT02 binary plus switched PWM output binary plus switched with diagnosis binary plus switched with diagnosis + protection
	0x14	Configuration OUT03	rw	USINT	2	0 = 0x00 2 = 0x02 4 = 0x04 15 = 0x0F 16 = 0x10	off Input OUT03 binary plus switched PWM output binary plus switched with diagnosis binary plus switched with diagnosis + protection
	0x15	Configuration OUT04	rw	USINT	2	0 = 0x00 2 = 0x02 4 = 0x04 15 = 0x0F 16 = 0x10	off Input OUT04 binary plus switched PWM output binary plus switched with diagnosis binary plus switched with diagnosis + protection

Index	S-Idx	Parameter name	Data type		Default	Details	
	0x16	Configuration OUT05	rw	USINT	2	0 = 0x00 2 = 0x02 4 = 0x04 15 = 0x0F 16 = 0x10	off Input OUT05 binary plus switched PWM output binary plus switched with diagnosis binary plus switched with diagnosis + protection
	0x17	Configuration OUT06	rw	USINT	2	0 = 0x00 2 = 0x02 4 = 0x04 15 = 0x0F 16 = 0x10	off Input OUT06 binary plus switched PWM output binary plus switched with diagnosis binary plus switched with diagnosis + protection
	0x18	Configuration OUT07	rw	USINT	2	0 = 0x00 2 = 0x02 4 = 0x04 15 = 0x0F 16 = 0x10	off Input OUT07 binary plus switched PWM output binary plus switched with diagnosis binary plus switched with diagnosis + protection
0x2000	0x19	Configuration OUT08	rw	USINT	2	0 = 0x00 2 = 0x02 4 = 0x04	off Input OUT08 binary plus switched PWM output + PWM output, voltage controlled
	0x1A	Configuration OUT09	rw	USINT	2	0 = 0x00 2 = 0x02 4 = 0x04	off Input OUT09 binary plus switched PWM output + PWM output, voltage controlled
0x2000	0x1B	Configuration OUT10	rw	USINT	2	0 = 0x00 2 = 0x02 4 = 0x04	off Input OUT10 binary plus switched PWM output
	0x1C	Configuration OUT11	rw	USINT	2	0 = 0x00 2 = 0x02 4 = 0x04	off Input OUT11 binary plus switched PWM output
0x2000	0x1D	Configuration OUT12	rw	USINT	2	0 = 0x00 2 = 0x02	off Input OUT12 binary plus switched
	0x1E	Configuration OUT13	rw	USINT	2	0 = 0x00 2 = 0x02	off Input OUT13 binary plus switched
	0x1F	Configuration OUT14	rw	USINT	2	0 = 0x00 2 = 0x02	off Input OUT14 binary plus switched
	0x20	Configuration OUT15	rw	USINT	2	0 = 0x00 2 = 0x02	off Input OUT15 binary plus switched
0x2001	0x0	PWM frequency	ro	USINT	12	Largest sub-index supported	
	0x1	PWM frequency OUT00	rw	UINT	100	20…250	OUT00 PWM frequency [Hz]
	0x2	PWM frequency OUT01	rw	UINT	100	20…250	OUT01 PWM frequency [Hz]
	0x3	PWM frequency OUT02	rw	UINT	100	20…250	OUT02 PWM frequency [Hz]
	0x4	PWM frequency OUT03	rw	UINT	100	20…250	OUT03 PWM frequency [Hz]
	0x5	PWM frequency OUT04	rw	UINT	100	20…250	OUT04 PWM frequency [Hz]
	0x6	PWM frequency OUT05	rw	UINT	100	20…250	OUT05 PWM frequency [Hz]
	0x7	PWM frequency OUT06	rw	UINT	100	20…250	OUT06 PWM frequency [Hz]
	0x8	PWM frequency OUT07	rw	UINT	100	20…250	OUT07 PWM frequency [Hz]
	0x9	PWM frequency OUT08	rw	UINT	100	20…250	OUT08 PWM frequency [Hz]
	0xA	PWM frequency OUT09	rw	UINT	100	20…250	OUT09 PWM frequency [Hz]
	0xB	PWM frequency OUT10	rw	UINT	100	20…250	OUT10 PWM frequency [Hz]
	0xC	PWM frequency OUT11	rw	UINT	100	20…250	OUT11 PWM frequency [Hz]
0x2002	0x0	Current value	ro	USINT	2	Largest sub-index supported	
	0x1	Current value OUT00	ro	UINT	0	0…2 000	OUT00 output current [mA]
	0x2	Current value OUT01	ro	UINT	0	0…2 000	OUT01 output current [mA]
0x2004	0x0	P-value	ro	USINT	2	Largest sub-index supported	

Index	S-Idx	Parameter name	Data type		Default	Details	
	0x1	P-value OUT00	rw	USINT	30	0…255	OUT00 P-value for current control
	0x2	P-value OUT01	rw	USINT	30	0…255	OUT01 P-value for current control
0x2005	0x0	I-value	ro	USINT	2	Largest sub-index supported	
	0x1	I-value OUT00	rw	USINT	20	0…255	OUT00 I-value for current control
	0x2	I-value OUT01	rw	USINT	20	0…255	OUT01 I-value for current control
0x2006	0x0	PWM dither frequency	ro	USINT	12	Largest sub-index supported	
	0x1	PWM dither frequency OUT00	rw	UINT	0	0…PWMfreq / 2	OUT00 PWM dither frequency [Hz]
	0x2	PWM dither frequency OUT01	rw	UINT	0	0…PWMfreq / 2	OUT01 PWM dither frequency [Hz]
	0x3	PWM dither frequency OUT02	rw	UINT	0	0…PWMfreq / 2	OUT02 PWM dither frequency [Hz]
	0x4	PWM dither frequency OUT03	rw	UINT	0	0…PWMfreq / 2	OUT03 PWM dither frequency [Hz]
	0x5	PWM dither frequency OUT04	rw	UINT	0	0…PWMfreq / 2	OUT04 PWM dither frequency [Hz]
	0x6	PWM dither frequency OUT05	rw	UINT	0	0…PWMfreq / 2	OUT05 PWM dither frequency [Hz]
	0x7	PWM dither frequency OUT06	rw	UINT	0	0…PWMfreq / 2	OUT06 PWM dither frequency [Hz]
	0x8	PWM dither frequency OUT07	rw	UINT	0	0…PWMfreq / 2	OUT07 PWM dither frequency [Hz]
	0x9	PWM dither frequency OUT08	rw	UINT	0	0…PWMfreq / 2	OUT08 PWM dither frequency [Hz]
	0xA	PWM dither frequency OUT09	rw	UINT	0	0…PWMfreq / 2	OUT09 PWM dither frequency [Hz]
	0xB	PWM dither frequency OUT10	rw	UINT	0	0…PWMfreq / 2	OUT10 PWM dither frequency [Hz]
	0xC	PWM dither frequency OUT11	rw	UINT	0	0…PWMfreq / 2	OUT11 PWM dither frequency [Hz]
0x2007	0x0	PWM dither value	ro	USINT	12	Largest sub-index supported	
	0x1	PWM dither value OUT00	rw	UINT	0	0…1 000	OUT00 PWM dither value [‰]
	0x2	PWM dither value OUT01	rw	UINT	0	0…1 000	OUT01 PWM dither value [‰]
	0x3	PWM dither value OUT02	rw	UINT	0	0…1 000	OUT02 PWM dither value [‰]
	0x4	PWM dither value OUT03	rw	UINT	0	0…1 000	OUT03 PWM dither value [‰]
	0x5	PWM dither value OUT04	rw	UINT	0	0…1 000	OUT04 PWM dither value [‰]
	0x6	PWM dither value OUT05	rw	UINT	0	0…1 000	OUT05 PWM dither value [‰]
	0x7	PWM dither value OUT06	rw	UINT	0	0…1 000	OUT06 PWM dither value [‰]
	0x8	PWM dither value OUT07	rw	UINT	0	0…1 000	OUT07 PWM dither value [‰]
	0x9	PWM dither value OUT08	rw	UINT	0	0…1 000	OUT08 PWM dither value [‰]
	0xA	PWM dither value OUT09	rw	UINT	0	0…1 000	OUT09 PWM dither value [‰]
	0xB	PWM dither value OUT10	rw	UINT	0	0…1 000	OUT10 PWM dither value [‰]
	0xC	PWM dither value OUT11	rw	UINT	0	0…1 000	OUT11 PWM dither value [‰]
0x2012	0x0	Period input	ro	USINT	4	Largest sub-index supported	
	0x1	Period duration IN12	ro	UDINT	0	IN12 period duration [µs]	
	0x2	Period duration IN13	ro	UDINT	0	IN13 period duration [µs]	
	0x3	Period duration IN14	ro	UDINT	0	IN14 period duration [µs]	
	0x4	Period duration IN15	ro	UDINT	0	IN15 period duration [µs]	
0x2013	0x0	Period input number of periods for average	ro	USINT	4	Largest sub-index supported	
	0x1	Number of periods IN12	rw	USINT	4	1…255	IN12 number of periods
	0x2	Number of periods IN13	rw	USINT	4	1…255	IN13 number of periods
	0x3	Number of periods IN14	rw	USINT	4	1…255	IN14 number of periods
	0x4	Number of periods IN15	rw	USINT	4	1…255	IN15 number of periods

Index	S-Idx	Parameter name	Data type		Default	Details	
0x2014	0x0	Period input – ratio value	ro	USINT	4	Largest sub-index supported	
	0x1	Period ratio value IN12	ro	UINT	0	0…1 000	IN12 marc-to-space ratio [‰]
	0x2	Period ratio value IN13	ro	UINT	0	0…1 000	IN13 marc-to-space ratio [‰]
	0x3	Period ratio value IN14	ro	UINT	0	0…1 000	IN14 marc-to-space ratio [‰]
	0x4	Period ratio value IN15	ro	UINT	0	0…1 000	IN15 marc-to-space ratio [‰]
0x2015	0x0	Frequency input	ro	USINT	4	Largest sub-index supported	
	0x1	Frequency IN12	ro	REAL	1	0…30 000	IN12 frequency [Hz]
	0x2	Frequency IN13	ro	REAL	1	0…30 000	IN13 frequency [Hz]
	0x3	Frequency IN14	ro	REAL	1	0…30 000	IN14 frequency [Hz]
	0x4	Frequency IN15	ro	REAL	1	0…30 000	IN15 frequency [Hz]
0x2016	0x0	Timebase	ro	USINT	4	Largest sub-index supported	
	0x1	Timebase IN12	rw	UINT	50	0…2 000	IN12 timebase [ms]
	0x2	Timebase IN12	rw	UINT	50	0…2 000	IN13 timebase [ms]
	0x3	Timebase IN12	rw	UINT	50	0…2 000	IN14 timebase [ms]
	0x4	Timebase IN12	rw	UINT	50	0…2 000	IN15 timebase [ms]
0x2020	0x0	Input – short to supply voltage	ro	USINT	2	Largest sub-index supported	
	0x1	Short to supply voltage IN00...IN07	ro	USINT	0	0 = normal 1 = short circuit	channels (bit coded) 0b---- ---X = IN00 0b---- --X- = IN01 0b---- -X-- = IN02 0b---- X--- = IN03 0b---X ---- = IN04 0b--X- ---- = IN05 0b-X-- ---- = IN06 0bX--- ---- = IN07
	0x2	Short to supply voltage IN08…IN11	ro	USINT	0	0 = normal 1 = short circuit	channels (bit coded) 0b---- ---X = IN08 0b---- --X- = IN09 0b---- -X-- = IN10 0b---- X--- = IN11 0b---X ---- = IN12 0b--X- ---- = IN13 0b-X-- ---- = IN14 0bX--- ---- = IN15
0x2021	0x0	Input – wire break	ro	USINT	2	Largest sub-index supported	
	0x1	Wire break IN00...IN07	ro	USINT	0	0 = normal 1 = wire break	channels (bit coded) 0b---- ---X = IN00 0b---- --X- = IN01 0b---- -X-- = IN02 0b---- X--- = IN03 0b---X ---- = IN04 0b--X- ---- = IN05 0b-X-- ---- = IN06 0bX--- ---- = IN07
	0x2	Wire break IN08…IN11	ro	USINT	0	0 = normal 1 = wire break	channels (bit coded) 0b---- ---X = IN08 0b---- --X- = IN09 0b---- -X-- = IN10 0b---- X--- = IN11 0b---X ---- = IN12 0b--X- ---- = IN13 0b-X-- ---- = IN14 0bX--- ---- = IN15
0x2022	0x0	Output – short circuit	ro	USINT	1	Largest sub-index supported	

Index	S-Idx	Parameter name	Data type		Default	Details	

	0x1	Short circuit OUT00...OUT07	ro	USINT	0	0 = normal
						1 = short circuit
							channels (bit coded)
							`0b---- ---X = OUT00`
							`0b---- --X- = OUT01`
							`0b---- -X-- = OUT02`
							`0b---- X--- = OUT03`
							`0b---X ---- = OUT04`
							`0b--X- ---- = OUT05`
							`0b-X-- ---- = OUT06`
							`0bX--- ---- = OUT07`

0x2023	0x0	Output – open circuit	ro	USINT	1	Largest sub-index supported	

	0x1	Open circuit OUT00...OUT07	ro	USINT	0	0 = normal
						1 = open circuit
							channels (bit coded)
							`0b---- ---X = OUT00`
							`0b---- --X- = OUT01`
							`0b---- -X-- = OUT02`
							`0b---- X--- = OUT03`
							`0b---X ---- = OUT04`
							`0b--X- ---- = OUT05`
							`0b-X-- ---- = OUT06`
							`0bX--- ---- = OUT07`

0x2024	0x0	Output – overload	ro	USINT	1	Largest sub-index supported	

	0x1	Overload OUT00...OUT01	ro	USINT	0	0 = normal
						1 = overload
							channels (bit coded)
							`0b---- ---X = OUT00`
							`0b---- --X- = OUT01`

0x2025	0x0	Input analog – overcurrent	ro	USINT	1	Largest sub-index supported	

	0x1	Overcurrent IN00...IN03	ro	USINT	0	0 = normal
						1 = overcurrent
							channels (bit coded)
							`0b---- ---X = IN00`
							`0b---- --X- = IN01`
							`0b---- -X-- = IN02`
							`0b---- X--- = IN03`

0x2030	0x0	Input resistor	ro	USINT	2	Largest sub-index supported	
	0x1	Resistance IN04	ro	UINT	0	0...30 000	IN04 resistance [Ohms]
	0x2	Resistance IN05	ro	UINT	0	0...30 000	IN05 resistance [Ohms]
0x2040	0x0	System supply voltage VBBS	ro	USINT	1	Largest sub-index supported	
	0x1	VBBS	ro	USINT	0	VBBS voltage [mV]	
0x2041	0x0	Output supply voltage	ro	USINT	2	Largest sub-index supported	
	0x1	VBB1	ro	UINT	0	VBB1 voltage [mV]	
	0x2	VBB2	ro	UINT	0	VBB2 voltage [mV]	
0x2050		Device temperature	ro	INT	0	temperature [°C]	
0x20F0		Node ID	rw	USINT	124	1...125	node ID [!] value(0x20F0) != value(20F1)
0x20F1		Node ID	rw	USINT	124	1...125	node ID [!] value(0x20F0) != value(20F1)
0x20F2		Baud rate	rw	USINT	–	baud rate [!] value(0x20F2) != value(20F3)	
0x20F3		Baud rate	rw	USINT	–	baud rate [!] value(0x20F2) != value(20F3)	
0x20F4		Autostart	rw	UINT	0	nicht benutzt	
0x6000	0x0	Binary input Largest sub-index supported	ro	USINT	0x02	Binär-Eingänge Größter unterstützter Sub-Index = 2	

Index	S-Idx	Parameter name	Data type		Default	Details

	0x1	Binary inputs IN00 - IN07	ro	USINT	0	Binär-Eingänge IN00...IN07
						`0b---- ---X = IN00`
						`0b---- --X- = IN01`
						`0b---- -X-- = IN02`
						`0b---- X--- = IN03`
						`0b---X ---- = IN04`
						`0b--X- ---- = IN05`
						`0b-X-- ---- = IN06`
						`0bX--- ---- = IN07`

	0x2	Binary inputs IN08 - IN15	ro	USINT	0	Binär-Eingänge IN08...IN15
						`0b---- ---X = IN08`
						`0b---- --X- = IN09`
						`0b---- -X-- = IN10`
						`0b---- X--- = IN11`
						`0b---X ---- = IN12`
						`0b--X- ---- = IN13`
						`0b-X-- ---- = IN14`
						`0bX--- ---- = IN15`

0x6200	0x0	binary output Largest sub-index supported	ro	USINT	0x02	Binär-Ausgänge Größter unterstützter Sub-Index = 2

	0x1	Binary outputs OUT00 - OUT07	wo	USINT	0	Binär-Ausgänge OUT00...OUT07
						`0b---- ---X = OUT00`
						`0b---- --X- = OUT01`
						`0b---- -X-- = OUT02`
						`0b---- X--- = OUT03`
						`0b---X ---- = OUT04`
						`0b--X- ---- = OUT05`
						`0b-X-- ---- = OUT06`
						`0bX--- ---- = OUT07`

	0x2	Binary outputs OUT08 - OUT15	wo	USINT	0	Binär-Ausgänge OUT08...OUT15
						`0b---- ---X = OUT08`
						`0b---- --X- = OUT09`
						`0b---- -X-- = OUT10`
						`0b---- X--- = OUT11`
						`0b---X ---- = OUT12`
						`0b--X- ---- = OUT13`
						`0b-X-- ---- = OUT14`
						`0bX--- ---- = OUT15`

0x6404	0x0	analog input Largest sub-index supported	ro	USINT	0x04	Analog-Eingänge Größter unterstützter Sub-Index = 4
	0x1	Analog input IN00	ro	UINT	--	Analogwert von Eingang IN00
	0x2	Analog input IN01	ro	UINT	--	Analogwert von Eingang IN01
	0x3	Analog input IN02	ro	UINT	--	Analogwert von Eingang IN02
	0x4	Analog input IN03	ro	UINT	--	Analogwert von Eingang IN03
0x6414	0x0	PWM output Largest sub-index supported	ro	USINT	0x12	PWM-Ausgänge Größter unterstützter Sub-Index = 12
	0x1	PWM output OUT00	wo	UINT	--	Wert für PWM-Ausgang OUT00
	0x2	PWM output OUT01	wo	UINT	--	Wert für PWM-Ausgang OUT01
	0x3	PWM output OUT02	wo	UINT	--	Wert für PWM-Ausgang OUT02
	0x4	PWM output OUT03	wo	UINT	--	Wert für PWM-Ausgang OUT03
	0x5	PWM output OUT04	wo	UINT	--	Wert für PWM-Ausgang OUT04
	0x6	PWM output OUT05	wo	UINT	--	Wert für PWM-Ausgang OUT05
	0x7	PWM output OUT06	wo	UINT	--	Wert für PWM-Ausgang OUT06
	0x8	PWM output OUT07	wo	UINT	--	Wert für PWM-Ausgang OUT07
	0x9	PWM output OUT08	wo	UINT	--	Wert für PWM-Ausgang OUT08

Index	S-Idx	Parameter name	Data type		Default	Details
	0xA	PWM output OUT09	wo	UINT	--	Wert für PWM-Ausgang OUT09
	0xB	PWM output OUT10	wo	UINT	--	Wert für PWM-Ausgang OUT10
	0xC	PWM output OUT11	wo	UINT	--	Wert für PWM-Ausgang OUT11

Legende:
Data type: ro = read only / rw = read and write / wo = write only

7.3.4 Betrieb des E/A-Moduls

Inhalt

16433

Eingänge: PDO-Mapping (E/A-Modul)

15968

Die folgende Tabelle enthält aus der Steuerungskonfiguration die folgenden Einträge:

- CAN-Input
- Send PDO-Mapping

Bit-Codierung:

```
0b---- ---X = IN00 (IN08)
...
0bX--- ---- = IN07 (IN15)
```

TX-PDO	Variable Typ	COB-ID = NodeID + ...	Bemerkung
1	USINT	0x180	Eingangsbyte 0 (IN00…IN07)
1	USINT	0x180	Eingangsbyte 1 (IN08…IN15)
1	USINT	0x180	Kurzschluss gegen VBBS am Eingang (IN00…IN07)
1	USINT	0x180	Kurzschluss gegen VBBS am Eingang (IN08…IN15)
1	USINT	0x180	Drahtbruch am Eingang (IN00…IN07)
1	USINT	0x180	Drahtbruch am Eingang (IN08…IN15)
1	USINT	0x180	Überstrom am Eingang (IN00…IN03)
2	UINT	0x280	Analogeingang IN00
2	UINT	0x280	Analogeingang IN01
2	UINT	0x280	Analogeingang IN02
2	UINT	0x280	Analogeingang IN03
3	UINT	0x380	Widerstandseingang IN04
3	UINT	0x380	Widerstandseingang IN05
3	UINT	0x380	Ausgangsstrom an OUT00
3	UINT	0x380	Ausgangsstrom an OUT01
4	UDINT	0x480	Periodendauer in [µs] an IN12
4	UDINT	0x480	Periodendauer in [µs] an IN13
5	UDINT	0x181	Periodendauer in [µs] an IN14
5	UDINT	0x181	Periodendauer in [µs] an IN15
6	UINT	0x281	Puls-/Periode-Verhältnis in [‰] an IN12
6	UINT	0x281	Puls-/Periode-Verhältnis in [‰] an IN13
6	UINT	0x281	Puls-/Periode-Verhältnis in [‰] an IN14
6	UINT	0x281	Puls-/Periode-Verhältnis in [‰] an IN15
7	USINT	0x381	Frequenz in [Hz] an IN12
7	REAL	0x381	Frequenz in [Hz] an IN13

TX-PDO	Variable Typ	COB-ID = NodeID + ...	Bemerkung
8	REAL	0x481	Frequenz in [Hz] an IN14
8	REAL	0x481	Frequenz in [Hz] an IN15
9	USINT	0x182	Kurzschluss am Ausgang (OUT00…OUT07)
9	USINT	0x182	Drahtbruch am Ausgang (OUT00…OUT07)
9	USINT	0x182	Überstrom am Ausgang (OUT00…OUT01)
10	UINT	0x282	Versorgungsspannung an VBBS in [mV]
10	UINT	0x282	Versorgungsspannung an VBB1 in [mV]
10	UINT	0x282	Versorgungsspannung an VBB2 in [mV]
10	UINT	0x282	Temperatur im Gerät

Ausgänge: PDO-Mapping (E/A-Modul)

15969

Die folgende Tabelle enthält aus der Steuerungskonfiguration die folgenden Einträge:
- CAN-Output
- Receive PDO-Mapping

Bit-Codierung:

```
0b---- ---X = OUT00 (OUT08)
...
0bX--- ---- = OUT07 (OUT15)
```

RX-PDO	Variable Typ	COB-ID = NodeID + ...	Bemerkung
1	USINT	0x200	Ausgangsbyte 0 (OUT00...OUT07)
1	USINT	0x200	Ausgangsbyte 1 (OUT08...OUT15)
2	UINT	0x300	PWM-Ausgang OUT00
2	UINT	0x300	PWM-Ausgang OUT01
2	UINT	0x300	PWM-Ausgang OUT02
2	UINT	0x300	PWM-Ausgang OUT03
3	UINT	0x400	PWM-Ausgang OUT04
3	UINT	0x400	PWM-Ausgang OUT05
3	UINT	0x400	PWM-Ausgang OUT06
3	UINT	0x400	PWM-Ausgang OUT07
4	UINT	0x500	PWM-Ausgang OUT08
4	UINT	0x500	PWM-Ausgang OUT09
4	UINT	0x500	PWM-Ausgang OUT10
4	UINT	0x500	PWM-Ausgang OUT11

7.3.5 Systemmerker für das integrierte E/A-Modul ExB01

Inhalt

16270

Systemmerker (E/A-Modul ExB01)

15957

Für das integrierte E/A-Modul des Geräts gibt es keine Systemmerker.
Die Rückmeldungen erfolgen über Process-Data-Objects (PDOs) über die EDS-Datei.
→ Kapitel *Eingänge: PDO-Mapping (E/A-Modul)* (→ Seite 299)

7.3.6 Fehlermeldungen für das E/A-Modul

Inhalt

15891

EMCY-Objekte

15981

Folgende Fehlercodes gemäß DSP-401 und DSP-301 werden unterstützt:

EMCY-Code	Error-Reg	Zusatz-Code	Beschreibung
0x6100	0x11	0x00	Internal Software Überlauf einer RX-Warteschlange z.B. Frequenz der RX-PDOs ist zu groß Reset nur extern über Eintrag in Index 0x1003 SubIdx 00
0x6101	0x11	0x00	Internal Software Überlauf einer TX-Warteschlange z.B. Gerät kommt nicht auf den Bus Reset nur extern über Eintrag in Index 0x1003 SubIdx 00
0x8100	0x11	0x00	Monitoring (Guarding Error) Für die Zeit "guard time" • "life time factor" wird kein guard objekt empfangen Reset bei erneuter Kommunikation
0x8200	0x11	0x00	Monitoring (Synch Error) Für "communication cycle" wird kein synch objekt empfangen Nur in OPEATIONAL Reset bei Synch-OBJ oder PREOP

CANopen sieht nicht vor, dass zwei gleiche EMCY-Objekte hintereinander abgesetzt werden.

SDOs Fehlermeldungen

15951

Folgende Meldungen werden im Fehlerfall erzeugt:

Index	SubIdx	Parameter name	Data type		Default	Details
0x1001		Error register	ro	USINT	0	Fehlerregister bitcodiert gemäß Profil 301 zulässige Werte: `0b0000 0000` = kein Fehler `0b0000 0001` = generic error `0b0001 0000` = communication error `0b1000 0000` = manufacturer specific
0x1003	0x0	Predefined error field Number of entries	rw	UDINT	0	Es wird eine Fehlerliste mit 4 Einträgen unterstützt
	0x1	Error history	ro	UDINT	0	Aufgetretener Fehler; codiert entsprechend EMCY-Liste der zuletzt aufgetretene Fehler steht jeweils in Sub-Index 1
	0x2	Error history	ro	UDINT	0	Aufgetretener Fehler; codiert entsprechend EMCY-Liste
	0x3	Error history	ro	UDINT	0	Aufgetretener Fehler; codiert entsprechend EMCY-Liste
	0x4	Error history	ro	UDINT	0	Aufgetretener Fehler; codiert entsprechend EMCY-Liste
	0x5	Error history	ro	UDINT	0	Aufgetretener Fehler; codiert entsprechend EMCY-Liste
0x2020	0x0	Input – short to supply voltage	ro	USINT	2	Largest sub-index supported

	0x1	Short to supply voltage
		IN00...IN07
			ro	USINT	0	0 = normal
						1 = short circuit
						channels (bit coded)
						`0b---- ---X = IN00`
						`0b---- --X- = IN01`
						`0b---- -X-- = IN02`
						`0b---- X--- = IN03`
						`0b---X ---- = IN04`
						`0b--X- ---- = IN05`
						`0b-X-- ---- = IN06`
						`0bX--- ---- = IN07`

	0x2	Short to supply voltage
		IN08...IN11
			ro	USINT	0	0 = normal
						1 = short circuit
						channels (bit coded)
						`0b---- ---X = IN08`
						`0b---- --X- = IN09`
						`0b---- -X-- = IN10`
						`0b---- X--- = IN11`
						`0b---X ---- = IN12`
						`0b--X- ---- = IN13`
						`0b-X-- ---- = IN14`
						`0bX--- ---- = IN15`

0x2021	0x0	Input – wire break	ro	USINT	2	Largest sub-index supported

	0x1	Wire break IN00...IN07	ro	USINT	0	0 = normal
						1 = wire break
						channels (bit coded)
						`0b---- ---X = IN00`
						`0b---- --X- = IN01`
						`0b---- -X-- = IN02`
						`0b---- X--- = IN03`
						`0b---X ---- = IN04`
						`0b--X- ---- = IN05`
						`0b-X-- ---- = IN06`
						`0bX--- ---- = IN07`

	0x2	Wire break IN08...IN11	ro	USINT	0	0 = normal
						1 = wire break
						channels (bit coded)
						`0b---- ---X = IN08`
						`0b---- --X- = IN09`
						`0b---- -X-- = IN10`
						`0b---- X--- = IN11`
						`0b---X ---- = IN12`
						`0b--X- ---- = IN13`
						`0b-X-- ---- = IN14`
						`0bX--- ---- = IN15`

0x2022	0x0	Output – short circuit	ro	USINT	1	Largest sub-index supported

Index	SubIdx	Parameter name	Data type		Default	Details	

	0x1	Short circuit OUT00...OUT07	ro	USINT	0	0 = normal
						1 = short circuit
							channels (bit coded)
							`0b---- ---X = OUT00`
							`0b---- --X- = OUT01`
							`0b---- -X-- = OUT02`
							`0b---- X--- = OUT03`
							`0b---X ---- = OUT04`
							`0b--X- ---- = OUT05`
							`0b-X-- ---- = OUT06`
							`0bX--- ---- = OUT07`

0x2023	0x0	Output – open circuit	ro	USINT	1	Largest sub-index supported	

	0x1	Open circuit OUT00...OUT07	ro	USINT	0	0 = normal
						1 = open circuit
							channels (bit coded)
							`0b---- ---X = OUT00`
							`0b---- --X- = OUT01`
							`0b---- -X-- = OUT02`
							`0b---- X--- = OUT03`
							`0b---X ---- = OUT04`
							`0b--X- ---- = OUT05`
							`0b-X-- ---- = OUT06`
							`0bX--- ---- = OUT07`

0x2024	0x0	Output – overload	ro	USINT	1	Largest sub-index supported	

	0x1	Overload OUT00...OUT01	ro	USINT	0	0 = normal
						1 = overload
							channels (bit coded)
							`0b---- ---X = OUT00`
							`0b---- --X- = OUT01`

0x2025	0x0	Input analog – overcurrent	ro	USINT	1	Largest sub-index supported	

	0x1	Overcurrent IN00...IN03	ro	USINT	0	0 = normal
						1 = overcurrent
							channels (bit coded)
							`0b---- ---X = IN00`
							`0b---- --X- = IN01`
							`0b---- -X-- = IN02`
							`0b---- X--- = IN03`

Legende
Data type: ro = read only / rw = read and write / wo = write only

7.4 Fehler-Tabellen

Inhalt

19606

7.4.1 Fehler-Codes

Inhalt

12334

Übersicht der Fehler-Codes, die von einigen Funktionsbausteinen ausgegeben werden.

Der 32-Bit-Fehler-Code besteht aus vier 8-Bit-Werten (DWORD).

4. Byte	3. Byte	2. Byte	1. Byte
Fehlerklasse	anwendungsspezifischer Fehler-Code	Fehlerquelle	Fehlerursache

Fehlerursache (1. Byte)

19273

Wert dez	hex	Beschreibung
0	00	keine Fehlerursache oder: anwendungsspezifischer Fehler
1	01	Bruch
2	02	Schluss
4	04	Überlast
5	05	Unterspannung
6	06	Überspannung
7	07	Stromregelung
24	18	Temperatur
25	19	Relaiskontakt
26	20	Speichertest
27	21	Adresstest
48	30	Interruptsystem
49	31	Zeitbasis
50	32	Befehlsausführung
51	33	Ganzzahl-Überlauf oder: Division durch Null
56	38	FPU Underflow
57	39	FPU Overflow
58	3A	FPU Division durch Null
59	3B	FPU unspezifischer Fehler
112	70	Kommunikation zum Co-Prozessor
128	80	CRC
129	81	Daten korrupt
130	82	Speicherschutz
131	83	keine Daten
144	90	Watchdog
145	91	Trap
147	93	Assertion fehlgeschlagen
194	C2	CAN Busoff
224	E0	Board Link Warnung (ExtendedController)
225	E1	Board Link Fehler (ExtendedController)
240	F0	Seriennummer
241	F1	Laufzeitsystem abgelaufen
248	F8	falscher Parameter

Fehlerquelle (2. Byte)

18660

Wert dez	hex	Beschreibung
0	00	keine Fehlerquelle oder: anwendungsspezifischer Fehler
1	01	CPU
2	02	Peripherie-Prozessor
3	03	Co-Prozessor
8	08	Floating-Point-Unit
16...31	10...1F	Eingang 0...15 (Standard-Seite)
32...63	20...3F	Eingang 0...31 (Extended-Seite)
64...79	40...4F	Ausgang 0...15 (Standard-Seite)
80..111	50...6F	Ausgang 0...31 (Extended-Seite)
128...131	80...83	CAN 1...4
144	90	Relaisspannung VBBO (Standard-Seite)
145	91	Relaisspannung VBBR (Standard-Seite)
146	92	VBBO (Standard-Seite)
147	93	VBBR (Standard-Seite)
148	94	VBBS (Standard-Seite)
149	95	Klemme 15
150	96	Relaisspannung VBB1 (Extended-Seite)
151	97	Relaisspannung VBB2 (Extended-Seite)
152	98	Relaisspannung VBB3 (Extended-Seite)
153	99	Relaisspannung VBB4 (Extended-Seite)
154	9A	VBBRel (Extended-Seite)
155	9B	VBB1 (Extended-Seite)
156	9C	VBB2 (Extended-Seite)
157	9D	VBB3 (Extended-Seite)
158	9E	VBB4 (Extended-Seite)
160	A0	Analog-Multiplexer
161	A1	Analog-Referenz
176	B0	internes Flash
177	B1	externes Flash
178	B2	internes RAM
179	B3	externes RAM
192	C0	Code Startupper
193	C1	Code Bootloader
194	C2	Code Laufzeitsystem
195	C3	Daten Peripherie-Prozessor
196	C4	Bootprojekt

Wert dez	hex	Beschreibung
197	C5	Code Anwendungsprogramm
198	C6	Scratch-Pad RAM
199	C7	Code Peripherie-Prozessor
224	E0	Systemdaten
225	E1	Systemeinstellungen
226	E2	Systeminformation
227	E3	Kalibrierdaten
228	E4	FRAM / MRAM (Anwenderbereich)

Anwendungsspezifischer Fehler-Code (3. Byte)

12338

Bei einem anwendungsspezifischen Fehler ist vorgeschrieben:
ERRORCODE Byte 1 = Fehlerursache = 0x00
ERRORCODE Byte 2 = Fehlerquelle = 0x00

► Anwendungsspezifische Fehler mit *ERROR_REPORT* der Steuerung melden.

Wert		Beschreibung
dez	hex	
0	00	kein anwendungsspezifischer Fehler
> 0	> 00	anwendungsspezifischer Fehler

Fehlerklasse (4. Byte)

19271

Diese Angaben gelten nur, wenn Eingang TEST = FALSE.

Wert		Beschreibung
dez	hex	
0	00	kein Fehler
1	01	allgemeiner Fehler, nicht sicherheitsrelevant > Fehlermerker, Fehler-Code > Fehler rücksetzen möglich
3	03	ERROR STOP > Fehler-Code > Anwendungsprogramm abgebrochen > sicherer Zustand ► Power-Off-On-Reset erforderlich
4	04	fataler Fehler > Fehler-Code > Anwendungsprogramm abgebrochen > sicherer Zustand > Gerät startet neu (Soft-Reset)

14025

Fatale Fehler sind in der Anwendung nur dann sichtbar, wenn beim Auftreten des Fehlers der TEST-Eingang bereits aktiv ist.

Normalerweise führen fatale Fehler zum STOP der Steuerung, jedoch nicht bei aktivem TEST-Eingang.

Fehler-Codes: Beispiele

19274

Byte 2	▼ Byte 1 ▼		
Fehlerquelle [hex]	Fehlerursache [hex]	Beschreibung	Funktionsbaustein
10...1F	01	Leiterbruch Ixx	INPUT_ANALOG
20...3F (Ex)	01	Leiterbruch Ixx_E	INPUT_ANALOG_E
40...4F	01	Leiterbruch Qxx	
50...6F (Ex)	01	Leiterbruch Qxx_E	
10...1F	02	Kurzschluss Ixx	INPUT_ANALOG
20...3F (Ex)	02	Kurzschluss Ixx_E	INPUT_ANALOG_E
40...4F	02	Kurzschluss Qxx	
50...6F (Ex)	02	Kurzschluss Qxx_E	
10...1F	04	Überstrom Ixx	INPUT_ANALOG
20...3F (Ex)	04	Überstrom Ixx_E	INPUT_ANALOG_E
40...4F	04	Überlast Qxx	
50...6F (Ex)	04	Überlast Qxx_E	
90	05	Unterspannung Relaisspannung VBBS	
91	05	Unterspannung Relaisspannung VBBR	
92	05	Unterspannung VBBO	
93	05	Unterspannung VBBR	
94	05	Unterspannung VBBS	
95	05	Unterspannung Klemme 15	
96...99 (Ex)	05	Unterspannung Relaisspannung VBBx	
9A (Ex)	05	Unterspannung VBBrel	
9B...9E (Ex)	05	Unterspannung VBBx	
90	06	Überspannung Relaisspannung VBBS	
91	06	Überspannung Relaisspannung VBBR	
92	06	Überspannung VBBO	
93	06	Überspannung VBBR	
94	06	Überspannung VBBS > 32 V	
94	06	Überspannung VBBS > 34 V	
95	06	Überspannung Klemme 15	
96...99 (Ex)	06	Überspannung Relaisspannung VBBx	
9A (Ex)	06	Überspannung VBBrel	
9B...9E (Ex)	06	Überspannung VBBx	
40...4F	07	Stromregelung Qxx	
50...6F (Ex)	07	Stromregelung Qxx_E	
10...1F	08 (safe)	Safety-Diagnose am Stromeingang	
10...1F	09 (safe)	Safety-Diagnose am Spannungseingang	
40...4F	0A (safe)	Safety-Diagnose am aktivierten Ausgang ("stuck at 1", Querschluss)	
10...17	0B (safe)	Safety-Diagnose am SafetySwitch	SAFETY_SWITCH
A0	0C (safe)	Analogwerte Ueberwachung /Multiplexer	SAFETY_SWITCH
40...4F	0D (safe)	Safety-Diagnose am deaktivierten Ausgang ("stuck at 1")	

Byte 2	▼ Byte 1 ▼		
Fehlerquelle [hex]	Fehlerursache [hex]	Beschreibung	Funktionsbaustein
00	18	Temperaturfehler	
90...91	19 (safe)	Kontaktfehler Relais VBBO / VBBR	
B3	20	Speichertest im RAM fehlgeschlagen	
E4	20	Speichertest im FRAM/MRAM fehlgeschlagen	
B3	21 (safe)	Adresstest im RAM fehlgeschlagen	
01	30 (safe)	Falscher / fehlender Interrupt	
01	31 (safe)	Fehler Zeitbasis CPU	
08	39	Floating-Point Overflow	
08	3A	Floating-Point Division durch 0	
08	3B	unspezifizierter Floating-Point-Fehler	
C2	80	Prüfsummenfehler im LZS-Code (ifm-Code)	
C3 (safe)	80	Prüfsummenfehler im PCP-Daten-RAM	
C4	80	Prüfsummenfehler im Bootprojekt	
C5	80	Prüfsummenfehler im Anwendungsprogramm-Code	
C6	80	Prüfsummenfehler im SP-RAM	
C7 (safe)	80	Prüfsummenfehler im PCP-Code-RAM	
E0	80	Prüfsummenfehler in den Systemdaten	
E1	80	Prüfsummenfehler in den Systemvariablen	
E2	80	Prüfsummenfehler in den Systemparametern	
E3	80	Prüfsummenfehler in den Kalibrierdaten	
B3	81	Defekte Daten im RAM	
E3	83	Kalibrierdaten korrupt	
04	92 (safe)	Safety-Core gestoppt	
05	92 (safe)	Fehler Safety-Code	
80...83	C1	CANx Warning	
80...83	C2	CANx Busoff	
80...83	C3 (safe)	CANsafety Empfangsfehler	
80...83	C4 (safe)	CANsafety Sendefehler	
80, 82	C5 (safe)	CANsafety-Konfiguration korrupt	
00 (Ex)	E1	Board-Link-Fehler	
00	F0	Fehler Seriennummer	
00	F1	Laufzeitsystem abgelaufen	
00	F8	Parameterfehler	Alle FBs mit ERROR Ausgang
10...1F	F8	Parameterfehler Ixx	INPUT_ANALOG SET_INPUT_MODE
20...3F (Ex)	F8	Parameterfehler Ixx_E	INPUT_ANALOG_E SET_INPUT_MODE_E
40...4F	F8	Parameterfehler Qxx	OUTPUT_ANALOG SET_OUTPUT_MODE
50...6F (Ex)	F8	Parameterfehler Qxx_E	OUTPUT_ANALOG_E SET_OUTPUT_MODE_E
05 (safe)	ErrorCode	Fehler Safety-Code	
06 (safe)	TrapID	Fehler Safety-Code Trap	

Legende:
(Ex) = gilt nur für ExtendedController
(safe) = gilt nur für SafetyController

Die resultierende Fehlerklasse (= Byte 4) ergibt sich aus dem Zusammenhang der Situation und Parametrierung. Byte 3 (anwendungsspezifischer Fehler-Code) ist hier immer = 0.

7.4.2 Fehlermerker

19608

→ Kapitel *Systemmerker* (→ Seite 221)

7.4.3 Fehler: CAN / CANopen

19610
19604

→ Systemhandbuch "Know-How ecomat*mobile*"
→ Kapitel *CAN / CANopen: Fehler und Fehlerbehandlung*

EMCY-Codes: CANx

13094

Die Angaben für CANx gelten für jede der CAN-Schnittstellen.

EMCY-Code Objekt 0x1003		Objekt 0x1001	herstellerspezifische Informationen					
Byte 0 [hex]	Byte 1 [hex]	Byte 2 [hex]	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Beschreibung
00	80	11	---	---	---	---	---	CANx Monitoring SYNC-Error (nur Slave)
00	81	11	---	---	---	---	---	CANx Warngrenze (> 96)
10	81	11	---	---	---	---	---	CANx Empfangspuffer Überlauf
11	81	11	---	---	---	---	---	CANx Sendepuffer Überlauf
30	81	11	---	---	---	---	---	CANx Guard-/Heartbeat-Error (nur Slave)

EMCY-Codes: E/As, System (Standard-Seite)

2668

Die folgenden EMCY-Meldungen werden automatisch versendet, wenn der FB *CANx_MASTER_EMCY_HANDLER* (→ Seite 84) zyklisch aufgerufen wird.

EMCY-Code Objekt 0x1003		Objekt 0x1001	herstellerspezifische Informationen					
Byte 0 [hex]	Byte 1 [hex]	Byte 2 [hex]	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Beschreibung
00	21	03	I07...I00	I15...I08				Leiterbruch Eingänge
08	21	03	I07...I00	I15...I08				Kurzschluss Eingänge
10	21	03	I07...I00	I15...I08				Überstrom 0...20 mA
00	23	03	Q07...Q00	Q15...Q08				Leiterbruch Ausgänge
08	23	03	Q07...Q00	Q15...Q08				Kurzschluss Ausgänge
00	31	05						Versorgungsspannung VBBS
00	33	05						Ausgangsspannung VBBO
08	33	05						Ausgangsspannung VBBR
00	42	09						Übertemperatur

8 Begriffe und Abkürzungen

A

Adresse

Das ist der „Name“ des Teilnehmers im Bus. Alle Teilnehmer benötigen eine unverwechselbare, eindeutige Adresse, damit der Austausch der Signale fehlerfrei funktioniert.

Anleitung

Übergeordnetes Wort für einen der folgenden Begriffe:
Montageanleitung, Datenblatt, Benutzerinformation, Bedienungsanleitung, Gerätehandbuch, Installationsanleitung, Onlinehilfe, Systemhandbuch, Programmierhandbuch, usw.

Anwendungsprogramm

Software, die speziell für die Anwendung vom Hersteller in die Maschine programmiert wird. Die Software enthält üblicherweise logische Sequenzen, Grenzwerte und Ausdrücke zum Steuern der entsprechenden Ein- und Ausgänge, Berechnungen und Entscheidungen.

Architektur

Spezifische Konfiguration von Hardware- und/oder Software-Elementen in einem System.

B

Baud

Baud, Abk.: Bd = Maßeinheit für die Geschwindigkeit bei der Datenübertragung. Baud ist nicht zu verwechseln mit "bits per second" (bps, Bit/s). Baud gibt zwar die Anzahl von Zustandsänderungen (Schritte, Takte) pro Sekunde auf einer Übertragungsstrecke an. Aber es ist nicht festgelegt, wie viele Bits pro Schritt übertragen werden. Der Name Baud geht auf den französischen Erfinder J. M. Baudot zurück, dessen Code für Telexgeräte verwendet wurde.
1 MBd = 1024 x 1024 Bd = 1 048 576 Bd

Bestimmungsgemäße Verwendung

Das ist die Verwendung eines Produkts in Übereinstimmung mit den in der Anleitung bereitgestellten Informationen.

Bootloader

Im Auslieferungszustand enthalten **ecomat*mobile***-Controller nur den Bootloader.
Der Bootloader ist ein Startprogramm, mit dem das Laufzeitsystem und das Anwendungsprogramm auf dem Gerät nachgeladen werden können.
Der Bootloader enthält Grundroutinen...

- zur Kommunikation der Hardware-Module untereinander,
- zum Nachladen des Laufzeitsystems.

Der Bootloader ist das erste Software-Modul, das im Gerät gespeichert sein muss.

Bus

Serielle Datenübertragung mehrerer Teilnehmer an derselben Leitung.

C

CAN

CAN = **C**ontroller **A**rea **N**etwork
CAN gilt als Feldbussystem für größere Datenmengen, das prioritätengesteuert arbeitet. Es gibt mehrere höhere Protokolle, die auf CAN aufsetzen, z. B. 'CANopen' oder 'J1939'.

CAN-Stack

CAN-Stack = Software-Komponente, die sich um die Verarbeitung von CAN-Telegramme kümmert.

CiA

CiA = CAN in Automation e.V.
Anwender- und Herstellerorganisation in Erlangen, Deutschland. Definitions- und Kontrollorgan für das CANopen-Protokoll.
Homepage → www.can-cia.org

CiA DS 304

DS = **D**raft **S**tandard
CANopen-Geräteprofil für sichere Kommunikation

CiA DS 401

DS = **D**raft **S**tandard
CANopen-Geräteprofil für digitale und analoge E/A-Baugruppen

CiA DS 402

DS = **D**raft **S**tandard
CANopen-Geräteprofil für Antriebe

CiA DS 403

DS = **D**raft **S**tandard
CANopen-Geräteprofil für Bediengeräte

CiA DS 404

DS = **D**raft **S**tandard
CANopen-Geräteprofil für Messtechnik und Regler

CiA DS 405

DS = **D**raft **S**tandard
CANopen-Spezifikation der Schnittstelle zu programmierbaren Steuerungen (IEC 61131-3)

CiA DS 406

DS = **D**raft **S**tandard
CANopen-Geräteprofil für Drehgeber / Encoder

CiA DS 407

DS = **D**raft **S**tandard
CANopen-Anwendungsprofil für den öffentlichen Nahverkehr

COB-ID

COB = **C**ommunication **Ob**ject = Kommunikationsobjekt
ID = **Id**entifier = Kennung
ID eines CANopen-Kommunikationsobjekts
Entspricht dem Identifier der CAN-Nachricht, mit der das Kommunikationsobjekt über den CAN-Bus gesendet wird.

CODESYS

CODESYS® ist eingetragene Marke der 3S – Smart Software Solutions GmbH, Deutschland.
'CODESYS for Automation Alliance™' vereinigt Firmen der Automatisierungsindustrie, deren Hardware-Geräte alle mit dem weit verbreiteten IEC 61131-3 Entwicklungswerkzeug CODESYS® programmiert werden.
Homepage → www.codesys.com

CSV-Datei

CSV = **C**omma **S**eparated **V**alues (auch: **C**haracter **S**eparated **V**alues)
Eine CSV-Datei ist eine Textdatei zur Speicherung oder zum Austausch einfach strukturierter Daten.
Die Dateinamen-Erweiterung lautet `.csv`.

Beispiel: Quell-Tabelle mit Zahlenwerten:

Wert 1.0	Wert 1.1	Wert 1.2	Wert 1.3
Wert 2.0	Wert 2.1	Wert 2.2	Wert 2.3
Wert 3.0	Wert 3.1	Wert 3.2	Wert 3.3

Daraus entsteht folgende CSV-Datei:

```
Wert 1.0;Wert 1.1;Wert 1.2;Wert 1.3
Wert 2.0;Wert 2.1;Wert 2.2;Wert 2.3
Wert 3.0;Wert 3.1;Wert 3.2;Wert 3.3
```

D

Datentyp

Abhängig vom Datentyp können unterschiedlich große Werte gespeichert werden.

Datentyp	min. Wert	max. Wert	Größe im Speicher
BOOL	FALSE	TRUE	8 Bit = 1 Byte
BYTE	0	255	8 Bit = 1 Byte
WORD	0	65 535	16 Bit = 2 Bytes
DWORD	0	4 294 967 295	32 Bit = 4 Bytes
SINT	-128	127	8 Bit = 1 Byte
USINT	0	255	8 Bit = 1 Byte
INT	-32 768	32 767	16 Bit = 2 Bytes
UINT	0	65 535	16 Bit = 2 Bytes
DINT	-2 147 483 648	2 147 483 647	32 Bit = 4 Bytes
UDINT	0	4 294 967 295	32 Bit = 4 Bytes
REAL	$-3{,}402823466 \cdot 10^{38}$	$3{,}402823466 \cdot 10^{38}$	32 Bit = 4 Bytes
ULINT	0	18 446 744 073 709 551 615	64 Bit = 8 Bytes
STRING			number of char. + 1

DC

Direct **C**urrent = Gleichstrom

Diagnose

Bei der Diagnose wird der "Gesundheitszustand" des Gerätes geprüft. Es soll festgestellt werden, ob und gegebenenfalls welche →Fehler im Gerät vorhanden sind.

Je nach Gerät können auch die Ein- und Ausgänge auf einwandfreie Funktion überwacht werden:
- Drahtbruch,
- Kurzschluss,
- Wert außerhalb des Sollbereichs.

Zur Diagnose können Konfigurations-Dateien herangezogen werden, die während des "normalen" Betriebs des Gerätes erzeugt wurden.
Der korrekte Start der Systemkomponenten wird während der Initialisierungs- und Startphase überwacht.
Zur weiteren Diagnose können auch Selbsttests durchgeführt werden.

Dither

to dither (engl.) = schwanken / zittern.
Dither ist ein Bestandteil der →PWM-Signale zum Ansteuern von Hydraulik-Ventilen. Für die elektromagnetischen Antriebe von Hydraulik-Ventilen hat sich herausgestellt, dass sich die Ventile viel besser regeln lassen, wenn das Steuersignal (PWM-Impulse) mit einer bestimmten Frequenz der PWM-Frequenz überlagert wird. Diese Dither-Frequenz muss ein ganzzahliger Teil der PWM-Frequenz sein.

DLC

Data **L**ength **C**ode = bei CANopen die Anzahl der Daten-Bytes in einer Nachricht.
Für →SDO: DLC = 8

DRAM

DRAM = **D**ynamic **R**andom **A**ccess **M**emory.
Technologie für einen elektronischen Speicherbaustein mit wahlfreiem Zugriff (Random Access Memory, RAM). Das speichernde Element ist dabei ein Kondensator, der entweder geladen oder entladen ist. Über einen Schalttransistor wird er zugänglich und entweder ausgelesen oder mit neuem Inhalt beschrieben. Der Speicherinhalt ist flüchtig: die gespeicherte Information geht bei fehlender Betriebsspannung oder zu später Wiederauffrischung verloren.

DTC

DTC = **D**iagnostic **T**rouble **C**ode = Fehler-Code
Beim Protokoll J1939 werden Störungen und Fehler über zugeordnete Nummern – den DTCs – verwaltet und gemeldet.

E

ECU

(1) **E**lectronic **C**ontrol **U**nit = Steuergerät oder Mikrocontroller
(2) **E**ngine **C**ontrol **U**nit = Steuergerät eines Motors

EDS-Datei

EDS = **E**lectronic **D**ata **S**heet = elektronisch hinterlegtes Datenblatt, z.B. für:
- Datei für das Objektverzeichnis im CANopen-Master,
- CANopen-Gerätebeschreibungen.

Via EDS können vereinfacht Geräte und Programme ihre Spezifikationen austauschen und gegenseitig berücksichtigen.

Embedded Software

System-Software, Grundprogramm im Gerät, praktisch das →Laufzeitsystem.
Die Firmware stellt die Verbindung her zwischen der Hardware des Gerätes und dem Anwendungsprogramm. Die Firmware wird vom Hersteller der Steuerung als Teil des Systems geliefert und kann vom Anwender nicht verändert werden.

EMCY

Abkürzung für Emergency (engl.) = Notfall
Nachricht im CANopen-Protokoll, mit der Fehler gemeldet werden.

EMV

EMV = **E**lektro-**M**agnetische **V**erträglichkeit.
Gemäß der EG-Richtlinie (2004/108/EG) zur elektromagnetischen Verträglichkeit (kurz EMV-Richtlinie) werden Anforderungen an die Fähigkeit von elektrischen und elektronischen Apparaten, Anlagen, Systemen oder Bauteilen gestellt, in der vorhandenen elektromagnetischen Umwelt zufriedenstellend zu arbeiten. Die Geräte dürfen ihre Umgebung nicht stören und dürfen sich von äußerlichen elektromagnetischen Störungen nicht ungünstig beeinflussen lassen.

Ethernet

Ethernet ist eine weit verbreitete, herstellerneutrale Netzwerktechnologie, mit der Daten mit einer Geschwindigkeit von 10 bis 10 000 Millionen Bit pro Sekunde (Mbps) übertragen werden können. Ethernet gehört zu der Familie der sogenannten „bestmöglichen Datenübermittlung“ auf einem nicht exklusiven Übertragungsmedium. 1972 entwickelt, wurde das Konzept 1985 als IEEE 802.3 spezifiziert.

EUC

EUC = **E**quipment **U**nder **C**ontrol (kontrollierte Einrichtung).
EUC ist eine Einrichtung, Maschine, Gerät oder Anlage, verwendet zur Fertigung, Stoffumformung, zum Transport, zu medizinischen oder anderen Tätigkeiten (→ IEC 61508-4, Abschnitt 3.2.3). Das EUC umfasst also alle Einrichtungen, Maschinen, Geräte oder Anlagen, die →Gefährdungen verursachen können und für die sicherheitsgerichtete Systeme erforderlich sind.
Falls eine vernünftigerweise vorhersehbare Aktivität oder Inaktivität zu durch das EUC verursachten Gefährdungen mit unvertretbarem Risiko führt, sind Sicherheitsfunktionen erforderlich, um einen sicheren Zustand für das EUC zu erreichen oder aufrecht zu erhalten. Diese Sicherheitsfunktionen werden durch ein oder mehrere sicherheitsgerichtete Systeme ausgeführt.

F

Fehlanwendung

Das ist die Verwendung eines Produkts in einer Weise, die vom Konstrukteur nicht vorgesehen ist.
Eine Fehlanwendung führt meist zu einer →Gefährdung von Personen oder Sachen.
Vor vernünftigerweise, vorhersehbaren Fehlanwendungen muss der Hersteller des Produkts in seinen Benutzerinformationen warnen.

FiFo

FIFO (**F**irst **I**n, **F**irst **O**ut) = Arbeitsweise des Stapelspeichers: Das Datenpaket, das zuerst in den Stapelspeicher geschrieben wurde, wird auch als erstes gelesen. Pro Identifier steht ein solcher Zwischenspeicher (als Warteschlange) zur Verfügung.

Flash-Speicher

Flash-ROM (oder Flash-EPROM oder Flash-Memory) kombiniert die Vorteile von Halbleiterspeicher und Festplatten. Die Daten werden allerdings wie bei einer Festplatte blockweise in Datenblöcken zu 64, 128, 256, 1024, ... Byte zugleich geschrieben und gelöscht.

Vorteile von Flash-Speicher

- Die gespeicherten Daten bleiben auch bei fehlender Versorgungsspannung erhalten.
- Wegen fehlender beweglicher Teile ist Flash geräuschlos, unempfindlich gegen Erschütterungen und magnetische Felder.

Nachteile von Flash-Speicher

- Begrenzte Zahl von Schreib- bzw. Löschvorgängen, die eine Speicherzelle vertragen kann:
  • Multi-Level-Cells: typ. 10 000 Zyklen
  • Single-Level-Cells: typ. 100 000 Zyklen
- Da ein Schreibvorgang Speicherblöcke zwischen 16 und 128 kByte gleichzeitig beschreibt, werden auch Speicherzellen beansprucht, die gar keiner Veränderung bedürfen.

FRAM

FRAM, oder auch FeRAM, bedeutet **Fe**rroelectric **R**andom **A**ccess **M**emory. Der Speicher- und Löschvorgang erfolgt durch eine Polarisationsänderung in einer ferroelektrischen Schicht.
Vorteile von FRAM gegenüber herkömmlichen Festwertspeichern:
• nicht flüchtig,
• kompatibel zu gängigen EEPROMs, jedoch:
• Zugriffszeit ca. 100 ns,
• fast unbegrenzt viele Zugriffszyklen möglich.

H

Heartbeat

Heartbeat (engl.) = Herzschlag.
Die Teilnehmer senden regelmäßig kurze Signale. So können die anderen Teilnehmer prüfen, ob ein Teilnehmer ausgefallen ist.

HMI

HMI = **H**uman **M**achine **I**nterface = Mensch-Maschine-Schnittstelle

I

ID – Identifier

ID = **Id**entifier = Kennung

Name zur Unterscheidung der an einem System angeschlossenen Geräte / Teilnehmer oder der zwischen den Teilnehmern ausgetauschten Nachrichtenpakete.

IEC 61131

Norm: Grundlagen Speicherprogrammierbarer Steuerungen
- Teil 1: Allgemeine Informationen
- Teil 2: Betriebsmittelanforderungen und Prüfungen
- Teil 3: Programmiersprachen
- Teil 5: Kommunikation
- Teil 7: Fuzzy-Control-Programmierung

IEC-User-Zyklus

IEC-User-Zyklus = SPS-Zyklus im CODESYS-Anwendungsprogramm.

IP-Adresse

IP = **I**nternet **P**rotocol = Internet-Protokoll.
Die IP-Adresse ist eine Nummer, die zur eindeutigen Identifizierung eines Internet-Teilnehmers notwendig ist. Zur besseren Übersicht wird die Nummer in 4 dezimalen Werten geschrieben, z. B. 127.215.205.156.

ISO 11898

Norm: Straßenfahrzeuge – CAN-Protokoll
- Teil 1: Bit-Übertragungsschicht und physikalische Zeichenabgabe
- Teil 2: High-speed medium access unit
- Teil 3: Fehlertolerante Schnittstelle für niedrige Geschwindigkeiten
- Teil 4: Zeitgesteuerte Kommunikation
- Teil 5: High-speed medium access unit with low-power mode

ISO 11992

Norm: Straßenfahrzeuge – Austausch von digitalen Informationen über elektrische Verbindungen zwischen Zugfahrzeugen und Anhängefahrzeugen
- Teil 1: Bit-Übertragungsschicht und Sicherungsschicht
- Teil 2: Anwendungsschicht für die Bremsausrüstung
- Teil 3: Anwendungsschicht für andere als die Bremsausrüstung
- Teil 4: Diagnose

ISO 16845

Norm: Straßenfahrzeuge – Steuergerätenetz (CAN) – Prüfplan zu Konformität

J

J1939

→ SAE J1939

K

Klemme 15

Klemme 15 ist in Fahrzeugen die vom Zündschloss geschaltete Plusleitung.

L

Laufzeitsystem

Grundprogramm im Gerät, stellt die Verbindung her zwischen der Hardware des Gerätes und dem Anwendungsprogramm.
→ Kapitel *Software-Module für das Gerät* (→ Seite 37)

LED

LED = **L**ight **E**mitting **D**iode = Licht aussendende Diode.
Leuchtdiode, auch Luminiszenzdiode, ein elektronisches Element mit hoher, farbiger Leuchtkraft auf kleinem Volumen bei vernachlässigbarer Verlustleistung.

Link

Ein Link ist ein Querverweis zu einer anderen Stelle im Dokument oder auf ein externes Dokument.

LSB

Least **S**ignificant **B**it/Byte = Niederwertigstes Bit/Byte in einer Reihe von Bit/Bytes.

M

MAC-ID

MAC = **M**anufacturer's **A**ddress **C**ode
= Hersteller-Seriennummer.
→ID = **Id**entifier = Kennung
Jede Netzwerkkarte verfügt über eine so genannte MAC-Adresse, ein unverwechselbarer, auf der ganzen Welt einzigartiger Zahlencode – quasi eine Art Seriennummer. So eine MAC-Adresse ist eine Aneinanderreihung von 6 Hexadezimalzahlen, etwa "00-0C-6E-D0-02-3F".

Master

Wickelt die komplette Organisation auf dem →Bus ab. Der Master entscheidet über den zeitlichen Buszugriff und fragt die →Slaves zyklisch ab.

MMI

MMI = **M**ensch-**M**aschine-**I**nterface
→ *HMI* (→ Seite 319)

MRAM

MRAM = **M**agnetoresistive **R**andom **A**ccess **M**emory
Die Informationen werden mit magnetischen Ladungselementen gespeichert. Dabei wird die Eigenschaft bestimmter Materialien ausgenutzt, die ihren elektrischen Widerstand unter dem Einfluss magnetischer Felder ändern.
Vorteile von MRAM gegenüber herkömmlichen Festwertspeichern:
- nicht flüchtig (wie FRAM), jedoch:
- Zugriffszeit nur ca. 35 ns,
- unbegrenzt viele Zugriffszyklen möglich.

MSB

Most **S**ignificant **B**it/Byte = Höchstwertiges Bit/Byte einer Reihe von Bits/Bytes.

N

NMT

NMT = **N**etwork **M**anagemen**t** = Netzwerk-Verwaltung (hier: im CANopen-Protokoll).
Der NMT-Master steuert die Betriebszustände der NMT-Slaves.

Node

Node (engl.) = Knoten. Damit ist ein Teilnehmer im Netzwerk gemeint.

Node Guarding

Node (engl.) = Knoten, hier: Netzwerkteilnehmer
Guarding (engl.) = Schutz
Parametrierbare, zyklische Überwachung von jedem entsprechend konfigurierten →Slave. Der →Master prüft, ob die Slaves rechtzeitig antworten. Die Slaves prüfen, ob der Master regelmäßig anfragt. Somit können ausgefallene Netzwerkteilnehmer schnell erkannt und gemeldet werden.

O

Obj / Objekt

Oberbegriff für austauschbare Daten / Botschaften innerhalb des CANopen-Netzwerks.

Objektverzeichnis

Das **Obj**ekt**v**erzeichnis OBV enthält alle CANopen-Kommunikationsparameter eines Gerätes, sowie gerätespezifische Parameter und Daten.

OBV

Das **Obj**ekt**v**erzeichnis OBV enthält alle CANopen-Kommunikationsparameter eines Gerätes, sowie gerätespezifische Parameter und Daten.

OPC

OPC = **O**LE for **P**rocess **C**ontrol = Objektverknüpfung und -einbettung für Prozesssteuerung
Standardisierte Software-Schnittstelle zur herstellerunabhängigen Kommunikation in der Automatisierungstechnik
OPC-Client (z.B. Gerät zum Parametrieren oder Programmieren) meldet sich nach dem Anschließen am OPC-Server (z.B. Automatisierungsgerät) automatisch bei diesem an und kommuniziert mit ihm.

operational

Operational (engl.) = betriebsbereit
Betriebszustand eines CANopen-Teilnehmers. In diesem Modus können →SDOs, →NMT-Kommandos und →PDOs übertragen werden.

P

PC-Karte

→ PCMCIA-Karte

PCMCIA-Karte

PCMCIA = Personal Computer Memory Card International Association, ein Standard für Erweiterungskarten mobiler Computer.
Seit der Einführung des Cardbus-Standards 1995 werden PCMCIA-Karten auch als PC-Karte (engl.: PC Card) bezeichnet.

PDM

PDM = **P**rocess and **D**ialog **M**odule = **P**rozess- und **D**ialog-**M**onitor.
Gerät zur Kommunikation des Bedieners mit der Maschine / Anlage.

PDO

PDO = **P**rocess **D**ata **O**bject = Nachrichten-Objekt mit Prozessdaten.
Die zeitkritischen Prozessdaten werden mit Hilfe der "Process Data Objects" (PDOs) übertragen. Die PDOs können beliebig zwischen den einzelnen Knoten ausgetauscht werden (PDO-Linking). Zusätzlich wird festgelegt, ob der Datenaustausch ereignisgesteuert (asynchron) oder synchronisiert erfolgen soll. Je nach der Art der zu übertragenden Daten kann die richtige Wahl der Übertragungsart zu einer erheblichen Entlastung des →CAN-Bus führen.
Dem Protokoll entsprechend, sind diese Dienste nicht bestätigte Dienste: es gibt keine Kontrolle, ob die Nachricht auch beim Empfänger ankommt. Netzwerkvariablen-Austausch entspricht einer "1-zu-n-Verbindung" (1 Sender zu n Empfängern).

PDU

PDU = **P**rotocol **D**ata **U**nit = Protokoll-Daten-Einheit.
Die PDU ist ein Begriff aus dem →CAN-Protokoll →SAE J1939. Sie bezeichnet einen Bestandteil der Ziel- oder Quelladresse.

PES

Programable **e**lectronic **s**ystem = Programmierbares elektronisches System ...
- zur Steuerung, zum Schutz oder zur Überwachung,
- auf der Basis einer oder mehrerer programmierbarer Geräte,
- einschließlich aller Elemente dieses Systems, wie Ein- und Ausgabegeräte.

PGN

PGN = **P**arameter **G**roup **N**umber = Parameter-Gruppennummer
PGN = PDU Format (PF) + PDU Source (PS)
Die Parameter-Gruppennummer ist ein Begriff aus dem →CAN-Protokoll →SAE J1939. Sie fasst die Teiladressen PF und PS zusammen.

PID-Regler

Der PID-Regler (proportional–integral–derivative controller) besteht aus folgenden Anteilen:
- P = Proportional-Anteil
- I = Integral-Anteil
- D = Differential-Anteil (jedoch nicht beim Controller CR04nn, CR253n).

Piktogramm

Piktogramme sind bildhafte Symbole, die eine Information durch vereinfachte grafische Darstellung vermitteln (→ Kapitel *Was bedeuten die Symbole und Formatierungen?* (→ Seite 7)).

Pre-Op

Pre-Op = PRE-OPERATIONAL mode (engl.) = Zustand vor 'betriebsbereit'.
Betriebszustand eines CANopen-Teilnehmers. Nach dem Einschalten der Versorgungsspannung geht jeder Teilnehmer automatisch in diesem Zustand. Im CANopen-Netz können in diesem Modus nur →SDOs und →NMT-Kommandos übertragen werden, jedoch keine Prozessdaten.

Prozessabbild

Mit Prozessabbild bezeichnet man den Zustand der Ein- und Ausgänge, mit denen die SPS innerhalb eines →Zyklusses arbeitet.

- Am Zyklus-Beginn liest die SPS die Zustände aller Eingänge in das Prozessabbild ein.
  Während des Zyklusses kann die SPS Änderungen an den Eingängen nicht erkennen.
- Im Laufe des Zyklusses werden die Ausgänge nur virtuell (im Prozessabbild) geändert.
- Am Zyklus-Ende schreibt die SPS die virtuellen Ausgangszustände auf die realen Ausgänge.

PWM

PWM = Puls-Weiten-Modulation
Bei dem PWM-Ausgangssignal handelt es sich um ein getaktetes Signal zwischen GND und Versorgungsspannung.
Innerhalb einer festen Periode (PWM-Frequenz) wird das Puls-/Pausenverhältnis variiert. Durch die angeschlossene Last stellt sich je nach Puls-/Pausenverhältnis der entsprechende Effektivstrom ein.

R

ratiometrisch

Ratio (lat.) = Verhältnis
Messungen können auch ratiometrisch erfolgen = Verhältnismessung. Wenn das Ausgangssinal eines Sensors proportional zu seiner Versorgungsspannung ist, kann durch ratiometrische Messung (= Messung im Verhältnis zur Versorgung) der Einfluss von Schwankungen der Versorgung reduziert, im Idealfall sogar beseitigt werden.
→ Analogeingang

RAW-CAN

RAW-CAN bezeichnet das reine →CAN-Protokoll, das ohne ein zusätzliches Kommunikationsprotokoll auf dem CAN-Bus (auf ISO/OSI-Schicht 2) arbeitet. Das CAN-Protokoll ist international nach →ISO 11898-1 definiert und garantiert zusätzlich in →ISO 16845 die Austauschbarkeit von CAN-Chips.

remanent

Remanente Daten sind gegen Datenverlust bei Spannungsausfall geschützt.
Z.B. kopiert das →Laufzeitsystem die remanenten Daten automatisch in einen →Flash-Speicher, sobald die Spannungsversorgung unter einen kritischen Wert sinkt. Bei Wiederkehr der Spannungsversorgung lädt das Laufzeitsystem die remanenten Daten zurück in den Arbeitsspeicher.
Dagegen sind die Daten im Arbeitsspeicher einer Steuerung flüchtig und bei Unterbrechung der Spannungsversorgung normalerweise verloren.

ro

ro = read only (engl.) = nur lesen
Unidirektionale Datenübertragung: Daten können nur gelesen werden, jedoch nicht verändert.

RTC

RTC = **R**eal **T**ime **C**lock = Echtzeituhr
Liefert (batteriegepuffert) aktuell Datum und Uhrzeit. Häufiger Einsatz beim Speichern von Fehlermeldungsprotokollen.

rw

rw = read/write (engl.) = lesen und schreiben
Bidirektionale Datenübertragung: Daten können sowohl gelesen als auch verändert werden.

S

SAE J1939

Das Netzwerkprotokoll SAE J1939 beschreibt die Kommunikation auf einem →CAN-Bus in Nutzfahrzeugen zur Übermittlung von Diagnosedaten (z.B.Motordrehzahl, Temperatur) und Steuerungsinformationen.
Norm: Recommended Practice for a Serial Control and Communications Vehicle Network

- Teil 2: Agricultural and Forestry Off-Road Machinery Control and Communication Network
- Teil 3: On Board Diagnostics Implementation Guide
- Teil 5: Marine Stern Drive and Inboard Spark-Ignition Engine On-Board Diagnostics Implementation Guide
- Teil 11: Physical Layer – 250 kBits/s, Shielded Twisted Pair
- Teil 13: Off-Board Diagnostic Connector
- Teil 15: Reduced Physical Layer, 250 kBits/s, Un-Shielded Twisted Pair (UTP)
- Teil 21: Data Link Layer
- Teil 31: Network Layer
- Teil 71: Vehicle Application Layer
- Teil 73: Application Layer – Diagnostics
- Teil 81: Network Management Protocol

SD-Card

Eine SD Memory Card (Kurzform für **S**ecure **D**igital Memory Card; deutsch: Sichere digitale Speicherkarte) ist ein digitales Speichermedium, das nach dem Prinzip der →Flash-Speicherung arbeitet.

SDO

SDO = **S**ervice **D**ata **O**bject = Nachrichten-Objekt mit Servicedaten.
Das SDO dient dem Zugriff auf Objekte in einem CANopen-Objektverzeichnis. Dabei fordern 'Clients' die gewünschten Daten von 'Servern' an. Die SDOs bestehen immer aus 8 Bytes.
Beispiele:

- Automatische Konfiguration aller →Slaves über SDOs beim Systemstart.
- Auslesen der Fehlernachrichten aus dem →Objektverzeichnis.

Jedes SDO wird auf Antwort überwacht und wiederholt, wenn sich innerhalb der Überwachungszeit der Slave nicht meldet.

Selbsttest

Testprogramm, das aktiv Komponenten oder Geräte testet. Das Programm wird durch den Anwender gestartet und dauert eine gewisse Zeit. Das Ergebnis davon ist ein Testprotokoll (Log-Datei), aus dem entnommen werden kann, was getestet wurde und ob das Ergebnis positiv oder negativ ist.

Slave

Passiver Teilnehmer am Bus, antwortet nur auf Anfrage des →Masters. Slaves haben im Bus eine eindeutige →Adresse.

Steuerungskonfiguration

Bestandteil der CODESYS-Bedienoberfläche.

► Programmierer teilt dem Programmiersystem mit, welche Hardware programmiert werden soll.
> CODESYS lädt die zugehörigen Bibliotheken.
> Lesen und schreiben der Peripherie-Zustände (Ein-/Ausgänge) ist möglich.

stopped

stopped (engl.) = angehalten
Betriebszustand eines CANopen-Teilnehmers. In diesem Modus werden nur →NMT-Kommandos übertragen.

Symbole

Piktogramme sind bildhafte Symbole, die eine Information durch vereinfachte grafische Darstellung vermitteln (→ Kapitel *Was bedeuten die Symbole und Formatierungen?* (→ Seite 7)).

Systemvariable

Variable, auf die via IEC-Adresse oder Symbolname aus der SPS zugegriffen werden kann.

T

Target

Das Target enthält für CODESYS die Hardware-Beschreibung des Zielgeräts, z.B.: Ein- und Ausgänge, Speicher, Dateiablageorte.
Entspricht einem elektronischen Datenblatt.

TCP

Das **T**ransmission **C**ontrol **P**rotocol ist Teil der Protokollfamilie TCP/IP. Jede TCP/IP-Datenverbindung hat einen Sender und einen Empfänger. Dieses Prinzip ist eine verbindungsorientierte Datenübertragung. In der TCP/IP-Protokollfamilie übernimmt TCP als verbindungsorientiertes Protokoll die Aufgabe der Datensicherheit, der Datenflusssteuerung und ergreift Maßnahmen bei einem Datenverlust. (vgl.: →UDP)

Template

Template (englisch = Schablone) ist eine Vorlage, die mit Inhalten gefüllt werden kann.
Hier: Eine Struktur von vorkonfigurierten Software-Elementen als Basis für ein Anwendungsprogramm.

U

UDP

UDP (**U**ser **D**atagram **P**rotocol) ist ein minimales, verbindungsloses Netzprotokoll, das zur Transportschicht der Internetprotokollfamilie gehört. Aufgabe von UDP ist es, Daten, die über das Internet übertragen werden, der richtigen Anwendung zukommen zu lassen.
Derzeit sind Netzwerkvariablen auf Basis von →CAN und UDP implementiert. Die Variablenwerte werden dabei auf der Basis von Broadcast-Nachrichten automatisch ausgetauscht. In UDP sind diese als Broadcast-Telegramme realisiert, in CAN als →PDOs.
Dem Protokoll entsprechend, sind diese Dienste nicht bestätigte Dienste: es gibt keine Kontrolle, ob die Nachricht auch beim Empfänger ankommt. Netzwerkvariablen-Austausch entspricht einer "1-zu-n-Verbindung" (1 Sender zu n Empfängern).

V

Verwendung, bestimmungsgemäß

Das ist die Verwendung eines Produkts in Übereinstimmung mit den in der Anleitung bereitgestellten Informationen.

W

Watchdog

Der Begriff Watchdog (englisch; Wachhund) wird verallgemeinert für eine Komponente eines Systems verwendet, die die Funktion anderer Komponenten beobachtet. Wird dabei eine mögliche Fehlfunktionen erkannt, so wird dies entweder signalisiert oder geeignete Programm-Verzweigungen eingeleitet. Das Signal oder die Verzweigungen dienen als Auslöser für andere kooperierende Systemkomponenten, die das Problem lösen sollen.

wo

wo = write only (engl.) = nur schreiben
Unidirektionale Datenübertragung: Daten können nur verändert werden, jedoch nicht gelesen.

Z

Zykluszeit

Das ist die Zeit für einen Zyklus. Das SPS-Programm läuft einmal komplett durch.

Je nach ereignisgesteuerten Verzweigungen im Programm kann dies unterschiedlich lange dauern.

9 Index

A

B

C

D

E

F

P

R

S

T

U

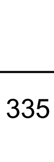
V

W

Z

10 Notizen • Notes • Notes





Notizen • Notes • Notes



11 ifm weltweit • ifm worldwide • ifm à l’échelle internationale

Stand: 2015-03-06 8310

www.ifm.com • E-Mail: info@ifm.com

Service-Hotline: 0800 16 16 16 4 (nur Deutschland, Mo...Fr, 07.00...18.00 Uhr)

ifm Niederlassungen • Sales offices • Agences

D	ifm electronic gmbh Vertrieb Deutschland Niederlassung Nord • 31135 Hildesheim • Tel. 0 51 21 / 76 67-0 Niederlassung West • 45128 Essen • Tel. 02 01 / 3 64 75 -0 Niederlassung Mitte-West • 58511 Lüdenscheid • Tel. 0 23 51 / 43 01-0 Niederlassung Süd-West • 64646 Heppenheim • Tel. 0 62 52 / 79 05-0 Niederlassung Baden-Württemberg • 73230 Kirchheim • Tel. 0 70 21 / 80 86-0 Niederlassung Bayern • 82178 Puchheim • Tel. 0 89 / 8 00 91-0 Niederlassung Ost • 07639 Tautenhain • Tel. 0 36 601 / 771-0 ifm electronic gmbh • Friedrichstraße 1 • 45128 Essen
A	ifm electronic gmbh • 1120 Wien • Tel. +43 16 17 45 00
AUS	ifm efector pty ltd. • Mulgrave Vic 3170 • Tel. +61 3 00 365 088
B, L	ifm electronic N.V. • 1731 Zellik • Tel. +32 2 / 4 81 02 20
BR	ifm electronic Ltda. • 03337-000, Sao Paulo SP • Tel. +55 11 / 2672-1730
CH	ifm electronic ag • 4 624 Härkingen • Tel. +41 62 / 388 80 30
CN	ifm electronic (Shanghai) Co. Ltd. • 201203 Shanghai • Tel. +86 21 / 3813 4800
CND	ifm efector Canada inc. • Oakville, Ontario L6K 3V3 • Tel. +1 800-441-8246
CZ	ifm electronic spol. s.r.o. • 25243 Průhonice • Tel. +420 267 990 211
DK	ifm electronic a/s • 2605 BROENDBY • Tel. +45 70 20 11 08
E	ifm electronic s.a. • 08820 El Prat de Llobregat • Tel. +34 93 479 30 80
F	ifm electronic s.a. • 93192 Noisy-le-Grand Cedex • Tél. +33 0820 22 30 01
FIN	ifm electronic oy • 00440 Helsinki • Tel . +358 75 329 5000
GB, IRL	ifm electronic Ltd. • Hampton, Middlesex TW12 2HD • Tel. +44 208 / 213-0000
GR	ifm electronic Monoprosopi E.P.E. • 15125 Amaroussio • Tel. +30 210 / 6180090
H	ifm electronic kft. • 9028 Györ • Tel. +36 96 / 518-397
I	ifm electronic s.a. • 20041 Agrate-Brianza (MI) • Tel. +39 039 / 68.99.982
IL	Astragal Ltd. • Azur 58001 • Tel. +972 3 -559 1660
IND	ifm electronic India Branch Office • Kolhapur, 416234 • Tel. +91 231-267 27 70
J	efector co., ltd. • Chiba-shi, Chiba 261-7118 • Tel. +81 043-299-2070
MAL	ifm electronic Pte. Ltd • 47100 Puchong Selangor • Tel. +603 8063 9522
MEX	ifm efector S. de R. L. de C. V. • Monterrey, N. L. 64630 • Tel. +52 81 8040-3535
N	Sivilingeniør J. F. Knudtzen A/S • 1396 Billingstad • Tel. +47 66 / 98 33 50
NL	ifm electronic b.v. • 3843 GA Harderwijk • Tel. +31 341 / 438 438
P	ifm electronic s.a. • 4410-136 São Félix da Marinha • Tel. +351 223 / 71 71 08
PL	ifm electronic Sp. z o.o. • 40-106 Katowice • Tel. +48 32-608 74 54
RA, ROU	ifm electronic s.r.l. • 1107 Buenos Aires • Tel. +54 11 / 5353 3436
ROK	ifm electronic Ltd. • 140-884 Seoul • Tel. +82 2 / 790 5610
RP	Gram Industrial, Inc. • 1770 Mantilupa City • Tel. +63 2 / 850 22 18
RUS	ifm electronic • 105318 Moscow • Tel. +7 495 921-44-14
S	ifm electronic a b • 41250 Göteborg • Tel. +46 31 / 750 23 00
SGP	ifm electronic Pte. Ltd. • Singapore 609 916 • Tel. +65 6562 8661/2/3
SK	ifm electronic s.r.o. • 835 54 Bratislava • Tel. +421 2 / 44 87 23 29
THA	SCM Allianze Co., Ltd. • Bangkok 10 400 • Tel. +66 02 615 4888
TR	ifm electronic Ltd. Sti. • 34381 Sisli/Istanbul • Tel. +90 212 / 210 50 80
UA	TOV ifm electronic • 02660 Kiev • Tel. +380 44 501 8543
USA	ifm efector inc. • Exton, PA 19341 • Tel. +1 610 / 5 24-2000
ZA	ifm electronic (Pty) Ltd. • 0157 Pretoria • Tel. +27 12 345 44 49